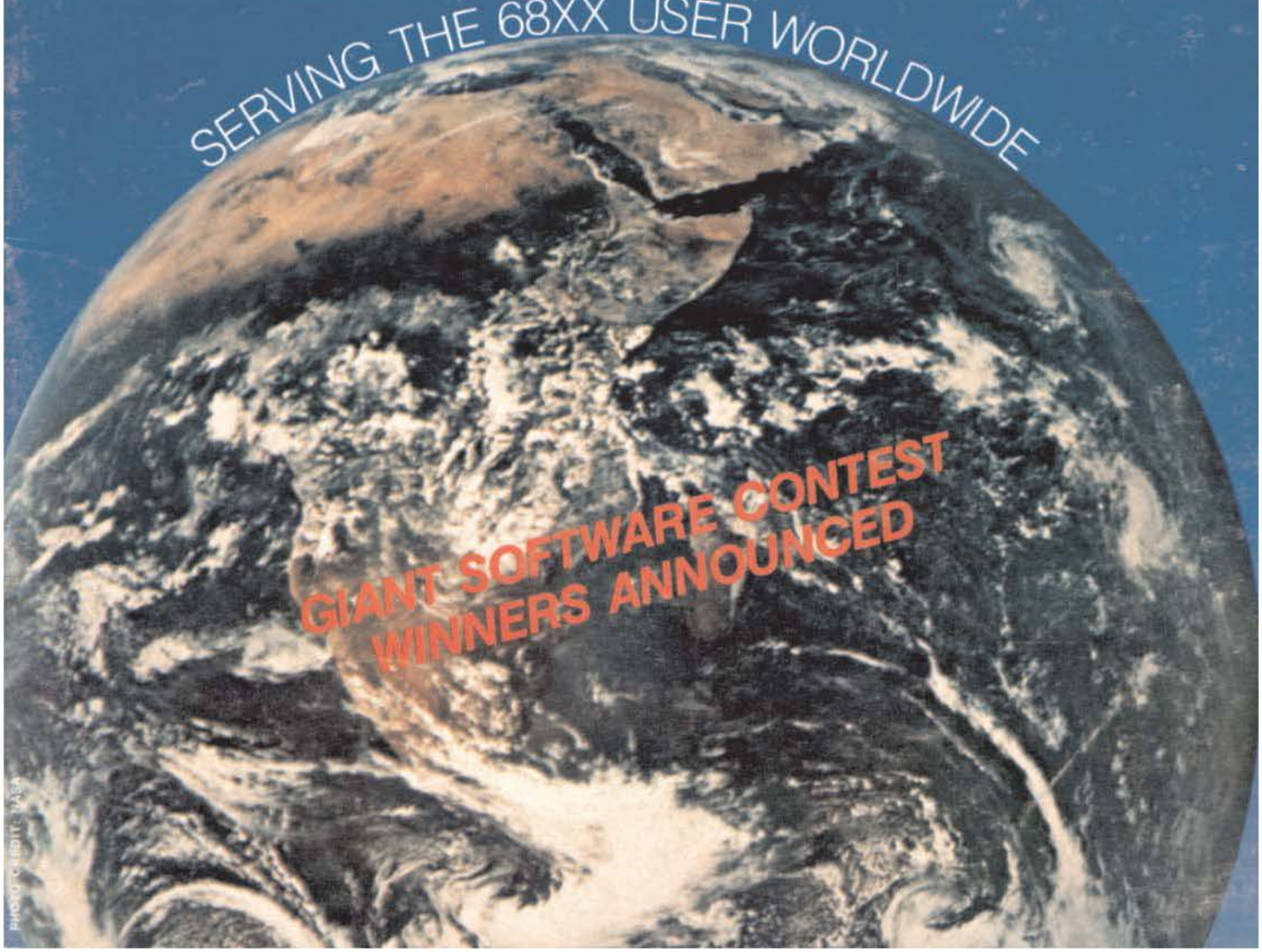
'68' MICRO JOURNAL

$2.50 USA

Australia A $ 4.00 New Zealand NZ $ 4.00
Singapore S $ 8.00 Hong Kong H $40.00
Malaysia M $ 8.00

VOLUME II ISSUE 12 • Devoted to the 68XX User • December 1980

"Small Computers Doing Big Things."



YOUR CHOICE-smart either way

- Over 140 software driven functions
- 82 x 24 or 82 x 20 screen format — software selectable
- High resolution 7 x 12 matrix characters — P-31 green phosphor
- Upper/lower case character set — plus graphics character set
- 56-key alphanumeric keyboard — plus 12-key cursor, numeric pad
- Internal editing functions — insert, delete, scroll, roll, slide, etc.
- Parallel printer I/O port
- 50 to 38,400 baud operation — programmable
- Cursor type, cursor position, print control characters, protected fields, shift inversion, dual intensity and many other features

8212 — twelve-inch diagnonal screen or **8209** — nine-inch diagnonal screen

SOUTHWEST TECHNICAL PRODUCTS CORPORATION
219 W. RHAPSODY
SAN ANTONIO, TEXAS 78216 (512) 344-0241

UniFLEX™

Multi-User

UniFLEX is the first full capability multi-user operating system available for microprocessors. Designed for the 6809 and 68000, it offers its users a very friendly computing environment. After a user 'logs-in' with his user name and password, any of the system programs may be run at will. One user may run the text editor while another runs BASIC and still another runs the C compiler. Each user operates in his own system environment, unaware of other user activity. The total number of users is only restricted by the resources and efficiency of the hardware in use.

Multi-Tasking

UniFLEX is a true multi-tasking operating system. Not only may several users run different programs, but one user may run several programs at a time. For example, a compilation of one file could be initiated while simultaneously making changes to another file using the text editor. New tasks are generated in the system by the 'fork' operation. Tasks may be run in the background or 'locked' in main memory to assist critical response times. Inter-task communication is also supported through the 'pipe' mechanism.

Support

The design of UniFLEX, with its hierarchical file system and device independent I/O, allows the creation of a variety of complex support programs. There is currently a wide variety of software available and under development. Included in this list is a Text Processing System for word processing functions, BASIC interpreter and precompiler for general programming and educational use, native C and Pascal compilers for more advanced programming, sort/merge for business applications, and a variety of debug packages. The standard system includes a text editor, assembler, and about forty utility programs. UniFLEX for 6809 is sold with a single CPU license and one years maintenance for $450.00. Additional yearly maintenance is available for $100.00. OEM licenses are also available.

FLEX™

UniFLEX is offered for the advanced microprocessor systems. FLEX, the industry standard for 6800 and 6809 systems, is offered for smaller, single user systems. A full line of FLEX support software and OEM licenses are also available.

technical systems consultants, inc.
Box 2570, West Lafayette, IN 47906
(317) 463-2502 Telex 276143

'68' MICRO JOURNAL

Portions of text prepared using the following.

SWTPC 6800-6809-DMAF2-COSI-CT82-Sprint 3
Southwest Technical Products
219 W. Rhapsody
San Antonio, Texas 78216

EDITOR - WORD PROCESSOR
Technical Systems Consultants, Inc.
Box 2573, W. Lafayette, IN 47906
FLEX is TM of TSC

GIMIX Super Mainframe-Assorted memory boards
GIMIX Inc.
1337 West 37th Place
Chicago, Il 60609

Publisher: Don Williams Sr.

Executive Editor: Larry Williams

Subscriptions and Office manager
Mary Robertson

General Girl 'Friday'
Joyce Williams

Contributing Editors:

Dr. Jack Bryant
Dr. Chuck Adams
Dr. Theo Elbert
Dr. Jeffery Brownstein
Dale Puckett
Russell Gore
Ron Anderson
John Jordon
Dennis Womack

* CONTENTS *

Send All Correspondence To:

'68' Micro Journal
3018 Hamill Rd.
PO Box 849
Hixson, Tennessee 37343

— Phone —
Office: 615-870-1993
Plant: 615-892-7544

'68' Micro Journal is published 12 times a year by '68' Micro Journal, 6131 Airways Blvd., Chattanooga, TN 37421. Second Class postage paid at Chattanooga, TN. Postmaster: Send Form 3579 to '68' Micro Journal, PO Box 849, Hixson, TN 37343.

1-Year $18.50 2-Year $32.50 3-Year $48.50

—ITEMS SUBMITTED FOR PUBLICATION—

(Letters to the Editor for Publication) All 'letters to the Editor' should be substantiated by facts. Opinions should be indicated as such. All letters must be signed. We are interested in receiving letters that will benefit or alert our readers. Praise as well as gripes is always good subject matter. Your name may be withheld upon request. If you have had a good experience with a 6800 vendor please put it in a letter. If the experience was bad put that in a letter also. Remember, if you tell us who they are then it is only fair that your name 'not' be withheld. This means that all letters published, of a critical nature, cannot have a name withheld. We will attempt to publish 'verbatim' letters that are composed using 'good taste.' We reserve the right to define (for '68' Micro) what constitutes 'good taste.'

(Articles and items submitted for publication) Please, always include your full name, address, and telephone number. Date and number all sheets. TYPE them if you can, poorly handwritten copy is sometimes the difference between go, no-go. All items should be on 8X11 inch, white paper. Most all art work will be reproduced photographically, this includes all listings, diagrams and other non-text material. All typewritten copy should be done with a NEW RIBBON. All hand drawn art should be black on white paper. Please no hand written code items over 50 bytes. Neatly typed copy will be directly reproduced. Column width should be 3¼ inches.

(Advertising) Any Classified: Maximum 20 words. All single letters and/or numbers will be considered one (1) word. No Commercial or Business Type Classified advertising. Classified ads will be published in our standard format. Classified ads $7.50 one time run, paid in advance.

Commercial and/or Business advertisers please write or phone for current rate sheet and publication lag time.

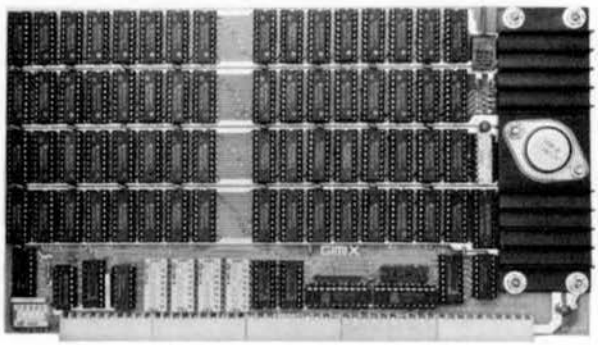
LOOK

WHAT'S COOKING on the FIFTY BUS

32K STATIC RAM BOARDS

Designed for use with:

★ **Existing SS50 Systems** ★ **SS50C Extended Address Systems**



- Assembled
- Burned In
- Tested

16K . . . $328.12

24K . . . $438.14

32K . . . $548.15

16K and 24K Versions are socketed for 32K and require only additional 2114's for expansion.

FEATURES:

- Decoding for 4 Extended Address Lines (allows memory decoding up to 1 megabyte)
- DIP-switch to set extended addressing or disable it
- 4 separate 8K blocks, addressable to any 8K boundary by DIP-switch
- Each 8K block may be individually disabled
- Write protect either of two 16K sections
- Low power consumption — uses 2114L low power RAMS
- Fully Socketed
- Gold Bus Connectors
- Guaranteed 2MHz operation

AND NOW . . . GIMIX OFFERS YOU A Choice of 6800 or 6809 CPU CARDS

You can order your system to fit your needs or select one of the below featured systems. Please contact the factory for further information and availability.

Add as much memory as you need using GIMIX Static RAM Cards for the utmost in reliability.

32K 6800 SYSTEM $1,694.59
Includes: Chassis, 6800 CPU, 32K RAM BOARD, I/O card

32K 6809 SYSTEM $1,844.69
Includes: Chassis, 6809 CPU, 32K RAM BOARD, I/O card

32K 6809 PLUS SYSTEM $1,994.79
Includes: Chassis, 32K RAM BOARD, I/O Card, and features our 6809 PLUS CPU Card with the Time of Day Clock option with battery back-up installed, as well as the 6840 Timer Package that provides 3 independent 16 bit counters.

This system also allows the following options to be added at additional cost:

- Battery back-up of the 1K RAM by substituting CMOS parts.
- A 9511 or 9512 Arithmetic Processor.
- GIMIX or SWTP Dynamic Address Translators.

EXPORT NOTES:

For 50Hz 230V C.V. POWER SUPPLY . . . . . Add $30.00
80 x 24 VIDEO BOARDS — Specify Format (No Added Charge)

On Orders under $250.00 for a Single Board, or Chips, please Add $30.00 Handling and we will ship Air Mail Prepaid. On all other orders we will ship via Emery Air Freight Collect, and we will charge no handling. All orders must be prepaid in U.S. Funds. Please note that foreign checks have been taking about eight weeks for collection, so we would advise wiring money or checks drawn on a bank account in the U.S. Our bank is the Continental Illinois National Bank of Chicago, Account #73-32033. Visa or Master Charge also accepted.

FACTORY PRIME STATIC RAMS

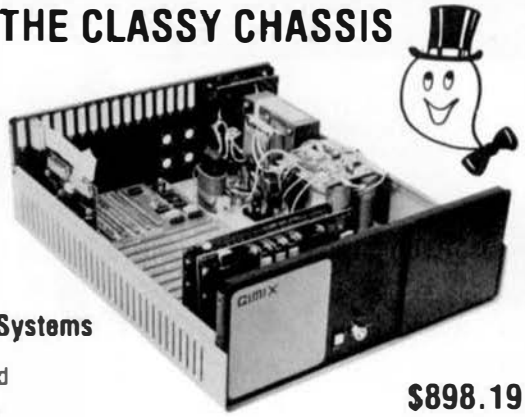
2114L 450 ns . . $5.90 300 ns . . $6.40 200 ns . . $6.90

4044 450 ns . . $5.90 250 ns . . $6.90

ADD $5.00 HANDLING ON ORDERS UNDER $200.00

GIMIX® and GHOST® are Registered Trademarks of GIMIX INC.

THE CLASSY CHASSIS



$898.19

- 25 amp (5V) terro-resonant constant voltage power supply.
- Heavy weight aluminum cabinet with 3 position key switch, fan, and provisions for two 5" disk drives.
- 6800/6809 Mother Board, fifteen 50 pin and eight DIP-switch addressable 30 pin slots (gold plated pins), fully decoded.
- Baud rate generator on I/O section of Mother Board.

I/O BOARDS

for the 30 PIN BUS:

1 Port Serial (RS 232 or 20MA, current loop)	$ 88.41
2 Port RS 232 Serial	128.43
2 Port Parallel	88.42

for the 50 PIN BUS:

8 Port RS 232 Serial	288.40
8 Port RS 232 Serial with on board Baud Rate generator.	318.46
8 Port Parallel	198.45

BOTH 6809 SYSTEMS FEATURE OUR NEW TERMINAL BASED GMXBUG 09 SYSTEM MONITOR

GMXBUG 09 includes advanced debugging tools, utility, and memory manipulation routines.

Both 6809 Systems:

- ★ Can be reconfigured to allow use of other system monitors (OS-9 and SBUG-E)
- ★ Include 1K of Scratchpad RAM on the CPU
- ★ Allow optional software switching of system monitors.

2MHz 6809's at slight additional cost when they become available.

Phone, write, or see your dealer for details and prices on our broad range of Boards and Systems for the SS50/SS50C bus and our AC Power Control Products for all computers.

GIMIX INC.

The Company that delivers Quality Electronic products since 1975.

1337 WEST 37th PLACE, CHICAGO, IL 60609
(312) 927-5510 • TWX 910-221-4055

See Ghost Ad Pages 36,39,40,48

INTRODUCING

6809 SOFTWARE POWER TOOLS

BY MICROWARE®

OS-9™ MULTIPROGRAMMING OPERATING SYSTEM

A true multitasking, real time operating system for timesharing, software development, database, process control, and other general applications. This versatile OS runs on almost any 6809-based computer.

- UNIX™ like file system with hierarchical directories, byte-addressable random-access files, and full file security.
- Versatile, easy-to-use input/output system is hardware independent and expandable to support almost any device with interrupt-driven, program-control, or BMA data transfer.
- Powerful "shell" command interpreter features: I/O redirection, multiple job stream processing, and more. Includes a comprehensive set of utility command programs.
- *OS-9 Level Two* uses hardware memory management and can address over one megabyte of memory. Also includes pipes and filters for inter-process data transfers.
- *OS-9 Level One* runs on systems without memory management hardware having up to 56K memory.

☐ OS-9 Level Two Operating System $495.00*
☐ OS-9 Level One Operating System $150.00*

STYLOGRAPH™ WORD PROCESSOR

A full-feature *screen-oriented* word processing system for interactive document entry and editing. Has cursor-oriented commands with dynamic screen refresh so the display is an exact image of the printed text.

- Editing commands include: line and character insertion and deletion; global string searching and replacement; "cut and paste" text rearrangement, etc.
- Formatting commands for: paging; left, right and center justification; headers and footers; line length and margins; and much more.
- I/O drivers for many popular video terminals including Hazeltine, ADDS, SWTPC, GIMIX and others.

☐ Standard version $135.00
☐ Special version for proportional printers $150.00

BASIC09™ PROGRAMMING LANGUAGE SYSTEM

Extended BASIC language compiler/interpreter with integrated text editor and debug package. Runs standard BASIC programs *or* minimally-modified PASCAL programs.

- Permits multiple named program modules having local variables and identifiers. Modules are reentrant, position independent and ROMable.
- Additional control statements for structured programming: IF ... THEN ... ELSE, FOR ... NEXT, REPEAT ... UNTIL, WHILE ... DO, LOOP ... ENDLOOP, EXITIF ... ENDEXIT.
- Allows user-defined data types and complex data structures. Five built-in data types: byte, integer, 9 digit floating-point, string and boolean.
- Outperforms any other BASIC on any 8-bit MPU.
- Available on ROM, disk or cassette tape. Runs under OS-9™ Level One or Level Two.

☐ Disk or tape $195.00*

OS-9™ TEXT EDITOR

Minimum-keystroke macro text editor useful for text preparation or interactive word processing.

- User-defined macros with parameters permit virtually unlimited command expansion. Macros can be saved, loaded and edited.
- Buffer, line and character oriented commands.
- Search, change and extend operations.
- Permits multiple input/output files.

☐ Disk or tape $75.00
☐ ROM set (2716) $90.00

OS-9™ INTERACTIVE ASSEMBLER

Compact Motorola compatible assembler for machine language program development.

- Operates in "batch" mode *or* interactive line-by-line mode.
- Facilities for generation of OS-9™ memory modules and system calls.
- Formatted listings include syntax *and* context error checking.
- Runs on OS-9™ Level One or Level Two.

☐ Disk or tape $75.00
☐ ROM set (2716) $90.00

OS-9™ INTERACTIVE DEBUGGER

Facilitates testing and debugging of machine- language programs.

- Includes common "monitor" functions: memory examine/change, breakpoints, display/change registers, etc.
- Calculator mode evaluates arithmetic expressions in hex, decimal or binary.
- Access to system commands.
- Available on ROM, disk or cassette tape.

☐ Disk or tape $35.00
☐ ROM (2716) $50.00

BASIC 09 is a trademark of Motorola. OS-9 is a trademark of Motorola and Microware®. UNIX is a trademark of Bell Telephone Laboratories.

Most software is available on ROM, diskette and tape in versions for many popular 6809 computers. Source listings and yearly maintenance/update service are sold separately for most programs.

*Specify manufacturer and type of CPU and I/O controllers. Contact Microware® for specific availability.

MICROWARE®

Microware Systems Corp., Dept. M3
5835 Grand Avenue
Des Moines, Iowa 50304
(515) 279-8844

INNOVATION AND PERFORMANCE

6800 SOFTWARE SUPER POWER

BY MICROWARE®

A/BASIC COMPILER

This BASIC compiler generates pure, fast, efficient 6800 machine language from easy to write BASIC source programs. Uses ultra-fast integer math, extended string functions, boolean operators and real-time operations. Output is ROMable and *runs without any run-time package.* Disk versions have disk I/O statements and require 12K memory and host DOS. Cassette version runs in 8K and requires RT/68 operating system.

☐ Disk Extended Version 2.1 SSB or FLEX* Diskette $150.00
☐ Cassette Version 1.0. $65.00

A/BASIC SOURCE GENERATOR

An "add-on" option for A/BASIC Compiler disk versions that adds an extra third pass which generates a full assembly-language output listing *and* assembly language source file. Uses original BASIC names and inserts BASIC source lines as comments.

☐ SSB or FLEX* Diskette $75.00

A/BASIC INTERPRETER

Here it is - a super-fast A/BASIC compiler! Now you can interactively edit, execute and debug A/BASIC programs with the ease of an interpreter - then compile to super efficient machine language. Also a superb stand-alone applications and control- oriented interpreter. Requires 8K RAM. The cassette version is perfect for Motorola ●2 kits.

☐ Cas tte, SSB or FLEX* Diskette $95.00

LISP INTERPRETER

The programming Language LISP offers exciting new possibilities for microcomputer applications. A highly interactive interpreter that uses list-type data structures which are simultaneously data and executable instructions. LISP features an unusual structured, recursive function-oriented syntax. Widely used for processing, artificial intelligence, education, simulation symbolic, and computer-aided design. 6800 LISP requires a minimum of 12K RAM.

☐ Cassette, SSB or FLEX* Diskette $75.00

*FLEX is a trademark of Technical Systems Consultants

RT/68 REAL TIME OPERATING SYSTEM

MIKBUG — compatible ROM that combines an improved monitor/debugger with a powerful multitasking real-time operating system. Supports up to 16 concurrent tasks at 8 priority levels plus real time clock and interrupt control. Thousands in use since 1976 handling all types of applications. Available on 6830 (MIKBUG-type) or 2708 (EPROM-type) ROM. Manual is a classic on 6800 real-time applications and contains a full source program listing.

☐ RT68MX (6830) $55.00
☐ RT68MXP (2708) $55.00

6800 CHESS

A challenging chess program for the 6800. Two selectable difficulty levels. Displays formatted chess boared on standard terminals. Requires 8K memory. Machine language with A/BASIC source listing.

☐ Cassette, SSB or FLEX* Diskette $50.00

Our software is available for most 6800 systems on cassette or diskette unless otherwise noted. Phone orders welcomed. We accept MASTERCHARGE and VISA. We try to ship orders within 24 hours of receipt. Please call or write if you require additional information or our free catalog. Microware® software is available for OEM and custom applications.

MICROWARE.

Microware Systems Corporation
P.O. Box 4865, Des Moines, IA 50304
(515) 279-8844

MICROSOFT DISK EXTENDED BASIC FOR OS-9™

Microsoft Basic is by far the world's most popular Basic language — and a vast library of applications software written for it can be used on your system: business packages, scientific, engineering and educational programs, games, etc. The OS-9™ version of Microsoft Basic is a full implementation of Release 5.0 with the following features:

- Four variable types: Integer, String, Single Precision Floating Point (7 Digit), Double Precision Floating Point (16 Digit).
- Full PRINT USING for formatted output (includes asterisk fill, floating $, scientific notation, trailing sign, comma insertion).
- Trace Facilities for program debugging.
- Extensive program editing facilities via EDIT command.
- Matrices with up to 255 dimensions.
- IF/THEN/ELSE and WHILE/WEND for structured programming.
- Automatic Line numbering and renumber.
- Dynamic string space allocation.
- Random and sequential file I/O with variable length records.
- Protected files can be saved in coded binary format.
- CHAIN and COMMON statements — programs may be linked together and share common variables.

This version of Microsoft Basic is not just a reassembled 6800 Basic — it has been enhanced to take full advantage of the 6809 and OS-9™ superior capabilities. It is also a reliable Basic that you can count on for your important programs.

☐ Microsoft Extended Basic Release 5.0 for OS-9™ $250.00

☐ **Also available:** Standard Microsoft 6800 or 6809 Basic Release 4.51 for Flex®. Many features of OS-9™ version. $250.00 *Trademark of Technical Systems Consultants

MICROWARE.

5835 Grand, Box 4865, Des Moines, IA 50304 • (515) 279-8844

SOFTWARE...

MSI has MORE!

WE INVITE YOU TO LOOK AT OUR NEW SOFTWARE CATALOG WHICH OFFERS NEW PROGRAMS FOR YOUR 6800 SYSTEM.

- All FLEX™ Programs from TSC are now available for MSI Computer Systems.
- MULTI-DISK FLEX™ from MSI allows the use of any combination of MSI disk devices to be used simultaneously, including the HD-8/R 10 megabyte drive.
- SORT/MERGE Program can be used manually or within other BASIC or assembler programs to perform high speed sorts of data files.
- Hemenway Associates Software Products for use under FLEX™ are available on the MSI System.
- TRS-80/MICROSOFT BASIC - MSI BASIC Translator allows MSI users to run the large library of basic programs written for the TRS-80 and other similar systems.
- SOFTWARE LIBRARY Programs keep track of all diskette and hard disk directories, giving alphabetical listings of available programs.
- SDOS Operating System.
- MULTI-USER/MULTI-TASKING SDOS Operating System allows any user to perform edits, assemblies, compilations, or program executions independently and simultaneously.
- All MSI software is supported on four (4) disk systems: quad density minifloppy, single and double density 8" floppy, as well hard disk systems.
- Complete BUSINESS APPLICATION PACKAGES including sales order entry, accounts receivable, inventory management, purchase order entry, accounts payable, and general ledger are available on MSI hard disk systems.
- PLOTTING PACKAGE gives daisy-wheel printers the capacity to perform graphics operations.
- LETTERWRITER Word Processing Software allows the use of daisy-wheel printers to generate documents and to handle correspondence automatically.

Send for your catalog today.

Midwest Scientific Instruments

220 W. Cedar • Olathe, Kansas 66061 • 913-764-3273

TWX 910 749 6403 (MSI OLAT)

Telex 42525 (MSI A OLAT)

As we enter the season of Thanksgiving, Christmas and New Year, it is our sincere prayer, that the Good Lord who made and substains us all, be with you and yours through the coming year. So to all of you from all of us

Merry Christmas and A VERY Happy New Year

RUMORS

Seems that more and more are coming along with 6809 machines. Rumor has it that Sony is serious about a 6809 machine to compete with the Radio Shack TRS80C™. We understand that it will be pointed at a higher level market and will be available with a choice of programming languages.

Also a very strong rumor that Cannon is coming soon with two new 6809 computers. One, the smaller, will retail for about $1,500, not much on this one. The other, with 64K and expandable to 96K plus. Both will be disk based but will run under Cannon's software.

Rumor has it fairly certain now that Apple is frantically working on a 68000 machine. Seems that they sure missed the mark when they attempted to upgrade with the 6502. Nobody (dealers and users) were very happy with the 'klug' called the Apple III, so I hear from many sources. About all they really inproved was software, and that sure could stand some improvement.

From PERCOM comes info that they are about to announce an adaptor board to interface the TRS80C™ to the S50 bus. We were told that a prototype is now up and running well. Should be available about the first of the year. Watch for our reports on this, could be tricky to work properly.

Last month I mentioned some potential problems with the TRS80C™ early delivery models. Seems that it has started already. Tandy, as of this writing, has a QC (quality control) hold on all TRS80C™ computers, so we are told by a store manager. We know of several that have failed in the first few hours. It is also rumored that a updated mask version of the 6809E is in the works. It will certainly be a joyful day when manufacturers stop using us, the users, as debugging tools.

DMW —

Computer Systems Developement Program

Disassemblers have been around a long time. Every assembly language programmer should have one in his tool kit of utilities. I would like to report on not just a disassembler, but on set of utilities. Bud Pass at Computer Systems Consultants, Inc. has a nifty package called the "CSC Program Development System." The package consists of the following:

1. CSC680XS - the disassembler
2. CSCSZAP - the binary file editor
3. CSCNAMES - replaces symbols with usual names
4. CSCXREF - the cross reference generator

These programs are intended to run under the FLEX™ operating system. Interestingly, they check to see whether they are running on a 6800 or on a 6809, and then automatically set up I/O vectors to either FLEX™ 1.0/2.0 or FLEX 9.0. The user is supplied with the source files for all programs. It is up to the user to assemble them on his 6800 or 6809.

These programs have a very friendly user interface. They prompt the user for required file names, etc. The disassembler and the binary file editor each have two levels of help commands. If the user has forgotten the code letters for the various commands (there are a lot), all he has to do is ask to see the menu! As the commands are used, the computer knows more and more about the program you are trying to "crack." If you, the user, have forgotten exactly what the computer thinks you want, just give the code for the status. See how friendly it is.

A disassembler takes raw machine code and converts it back to assembly language. Some disassmblers only work on programs in memory, but this one will dissassemble from memory or disk to disk. This means any program you have on disk can be disassembled and the output directed to a file on disk. You can then come back and edit and re-assemble the file. "What's this good for?" you ask. If the answer is not obvious, I'll give you some examples later on. This program will disassemble 6800, 6801, or 6809 code. All you have to do is specify which the code is, if different from the machine it is running on.

The binary file editor is called SUPER-ZAP. It will let you examine and modify a binary file. Its usefullness is in patching a program you don't have the source on. Portions of a program can be disassembled to the terminal, so you can see where to put your patch. SUPER-ZAP has all the flexibility of the disassembler.

The name changer program (CSCNAMES) will work with any text file. CSCNAMES will change every occurance of a text string to another text string. The effect is the same as the change command in the TSC Editor, except that the whole file is affected. The changes are not entered one at a time, but rather, are made into a text file. CSCNAMES will prompt you for the name of this file.

The disassembler normally assigns labels like "Lxxxx", where the "x" is a hex number. "xxxx" is the location referenced. For instance, "LC003" would be WARMS, the FLEX™ 9.0 warm start entry point. CSCNAMES can convert all the labels for locations you have identified into names you recognize. Bud Pass has been kind enough to supply change files for several versions of the FLEX™ operating system. These come in handy when you are disassembling a program that interfaces with FLEX™.

The cross reference generator (CSCXREF) will process a program source file and output an alphabetized cross reference. In the output, each symbol used is listed with the line number where it is used. Each line that references that symbol is also listed. The cross reference can be of great help in finding out: "How in the world did I wind up here?!"

Here's my evaluation: The programs work as intended and as advertised. Initially, there were a few bugs, but these have been cleared up. Bud Pass has

supported this software well by notifying users of problems and their solutions. This is much more than can be said of some software houses. As of this writing, I know of no other problem with it. Having spent several hours using the software, I think it is a very useful package. The price of the entire package is $50. Using it once or twice can make it pay for itself.

Following are the examples I promised:

control work with the 6800. I need a cross assembler that runs on the 6809 but accepts 6800 programs and generates 6800 machine language. No real problem, just disassemble that old 6800 assembler and re-assemble on the 6809. Oh, it will take several passes at identifying instruction and data areas in the old assembler before I have a working cross assembler. But, I'll bet I couldn't write a set of macros for the 6800 that fast.

Now that I have a 6800 cross assembler, I need a way to test the programs I generate, when developed on my 6809. TSC's Debug for the 6800 sure is nice, but now I'm running the 6809. I wish the 6800 Debug would run on the 6809.....

OS-9

"PUTTING THE 6809 TO WORK: A LOOK AT OS-9"

Many 6800 enthusiasts could barely contain themselves waiting for the introduction of the 6809 and the new generation of hardware and software that was expected to accompany it. It did finally arrive, and many 6809 systems and upgrades were sold, but not without some disappointment due to the lack of performance of the mostly rehashed 6800 software. For over a year, this is all that was available.

Well, the time has finally arrived, and some excellent 6809 software is finally becoming available from a number of sources. An important and instantly popular new software family is built around a UNIX-like operating system called "OS-9", and a powerful programming language called BASIC09. Both programs have many features that microcomputerists have generally not had available or even been exposed to before, so it's impossible to cover everything in detail. The object here is to give you an overview only: the best way to learn about the ins and outs of OS-9 is to read the manuals.

OS-9 is a hardware independent, multitasking, real-time operating designed from the ground up specifically for the 6809. The preceding mouthful means that it can run on almost any kind of 6809; that it can run more than one program at a time; and switch programs in response to "real-world" stimulus. These capabilities have obvious importance to persons who need systems that can do timesharing, process control, etc. The advantage to the small-system owner is not as apparent but it is very significant. Why? How many times have you wasted time waiting for a long program to run during which your computer was "dead" for all practical purposes? With a multitasking operating system your computer can run more than one program at the same time. This means you can play chess, run a long sort, print a letter, and monitor a security system: and ALL AT THE SAME TIME, even on a fairly small-sized system.

Is it realistic to expect your relatively inexpensive 32K 6809 computer to act like a big machine? You bet! In fact the 6809 has consistantly out-benchmarked the DEC PDP-11/23 minicomputer, and the 6809/OS-9/BASIC09 trio running multiuser applications will hold its own against the PDP-11/34 running DEC Basic-Plus using the RSTS-E timesharing system.

OS-9's design borrows many of the concepts pioneered by Bell Telephone Laboratory's respected UNIX operating system. There are many similarities in the file system works and in the syntax of the command interpreter (called the "shell"). But the similarity is mostly superficial because OS-9's internal workings and implementation is considerably different. Some of the software technology incorporated in OS-9 was not even developed in 1970 when UNIX was written. Other features such as the logical memory module system are the result of original research done specifically for the 6809 and OS-9.

We have just touched on the secret of OS-9's extraordinary versatility: modular software. Unlike most operating systems it is not one giant, monolithic program. Instead, it is a collection of a few basic and (usually) many optional, independent modules which are selected to fit a specific hardware arrangement and provide the desired functions. The "glue" that binds everything together is the "kernel" which is a package of routines that perform the essential basic functions. The kernel's size is about 3K and it resides in ROM memory at the highest memory addresses (OS-9 does not use a traditional monitor ROM). The kernel usually fills part of two 2K ROMs. The extra 1K of ROM space contains a device driver module for a disk or tape system, and also doubles as a bootstrap to load various other OS-9 modules into about 4K to 6K of RAM memory. It is possible to put these other modules in ROM if desired. The major functions of the kernel are: memory management, service request processing, CPU time management, and task switching.

Input and output processing is an important function of every operating system. OS-9 has a file management system based on the "unified I/O" concept which makes all I/O devices look like named files. For example, your system's printer is accessed as a "file" having a name like "printer", your terminal is a file named "term", etc. OS-9 automatically takes care of actual hardware-dependencies using software interfaces.

OS-9 has only one file "type": byte-addressable, random-access files. They can be read or written sequentially, or randomly starting at any byte address in the file. File names can be up to 29 characters long. Each file has an associated "owner" and attributes that control whether or not the file can be read, written or executed, by either the file's owner or by other users.

The disk allocation scheme was designed to perform data transfers at the maximum possible physical data rate for most devices. The maximum file size is over a trillion bytes, so multiple hard disks can be easily accomodated. It is simple to add drivers for a virtually unlimited number and different kinds of input/output devices. This does not present any software problems because each device is assigned a unique name.

The most visible part of the system is a program called the "shell". It is the command interpreter that asks for command lines from a terminal and executes them. In single-terminal systems the shell is automatically entered on system start-up; on multi-user systems a log-on program runs before the shell to ask for and verify a user name and password. The log-on program can also automatically select each user's own working disk directories (OS-9 has a hierarchical directory system) before entering the shell.

The shell is simultaneously easy-to-use and powerful because simple commands can make complicated things happen. For example, let's look at how OS-9 lets you run several programs at once. A command called "dir" prints out the file names on a directory. If you enter the command:

dir

the command (which is actually an independent program) runs, and the shell prints a prompt after it's done. But if you put an ampersand character at the end of a line, the command is executed as a separate, concurrent task and the shell can accept another command almost immediately, even before the command has finished. But you could get into trouble because more than one program can output lines of text to your terminal in random order. This can be very confusing.

The solution is a technique called "redirected input and output". Simply stated, it lets you change where a program's input and output data comes from or goes to, in the same command that calls the program. The "<" operator redirects the program's input from the file or device name that follows it, and the ">" operator likewise redirects the output. They can be used individually, or together. As an example, to run the "dir" command as a concurrent task with its output redirected from the terminal to the printer, the following command could be used:

dir >/printer &

The shell can also handle simultaneous execution of batch job streams (procedure files), multiple commands per line with sequential and concurrent execution operators, specific memory size assignments, inter-process communication files ("pipelines" and "filters"), and more.

Programmers who like to work at the machine language level will be suprised how easy it is to write assembly-language programs to run on OS-9. Even though the system calls have very powerful capabilites, they are easy to understand and NEVER use complicated data structures directly. You can forget about allocating and initializing buffers, "file control blocks", etc. OS-9 takes care of them internally so you never have to contend with them. Other features at this level include a single set of I/O service calls that work on any file or device; interprocess control and communications calls; automatic memory management; full support for moduler, ROMed, position-independent, reentrant code; a seperate "OS-9 System Programmer's Manual" with detailed descriptions of all system functions; and the ability to buy source code for the critical portions of the OS-9 I/O management system.

What does it take to run OS-9? Versions are available for almost all popular 6809-based computers including GIMIX, Southwest Technical Products, Smoke Signal Broadcasting, Motorola, Ackerman Digital Systems, Percom, Creative Micro Systems, and others. OS-9 can be purchased directly from many of these manufacturers or from Microware.

There are two basic versions: Level One and Level Two. The Level One version is for computers having from 8K to 56K of RAM memory. The Level Two version is intended for systems having up to a megabyte of RAM and having memory management/extended addressing hardware. Level Two also has the interprocess "pipe" file system as a standard feature (it's an option on Level One). Both versions are otherwise compatible, and either one can be used for timesharing applications. One requirement for either version is a real-time clock. Many 6809 CPU cards automatically come equipped with them. If not you have to buy one (such as the Southwest Technical Products MP-T), or make one from using a PIA or similar interface.

An operating system by itself is useless. Fortunately there is a good selection of support software available from Microware and other software houses. Much of the emphasis in OS-9 family software is on interactive, rather than batch, operation. The programs presently available include BASIC09, Microsoft Basic, assembler, editor, Stylograph screen-oriented word processor, an interactive machine-language debugger, and a wide assortment of various I/O expansion modules. Coming soon is a C compiler, A/Basic, Forth, and a database management system.

Judging by the number of products the 6809 is being designed into, it looks like it has the potential to capture a large part of the 8-bit microcomputer market within a couple of years. Even though the '09 is superior to its competition in almost every respect, it can't win without top-notch software. You can't write good software without good tools, and the OS-9 software family is an excellent set of tools.

Ken Kaplan, Microware.

UNIX is a register trademark of Bell Labratories.

UNIFLEX

The following article is copyrighted material and is published by permission of Technical System Consultants, Inc.

The UniFLEX™ Operating System

I. Introduction

This document provides an overview of the UniFLEX™ Operating System (UniFLEX is a trademark of Technical Systems Consultants, Inc.). Several of the important system features are described including a look at the user interface, the file system, and the program environment. Several terms will be defined so that those not completely familiar with modern computer terminology may follow. At the end of this document, some hints of the system's efficiency are provided.

II. System Description

UniFLEX™ is the first full featured time-sharing system available for use on microcomputers. It provides both a multi-user and a multi-tasking computing environment. Its design was influenced by two previous operating systems, FLEX™ and UNIX™ (FLEX is a trademark of Technical Systems Consultants, Inc., and UNIX is a trademark of Bell Laboratories). FLEX™ has been a very popular single-user operating system. It has proven to be quite reliable and extremely flexible in a wide variety of applications. UNIX, on the other hand, has been one of the most popular multi-user operating systems ever created. To this date, it has only been available for minicomputers. Its size has not made it a good candidate for micros, but limited feature versions will probably be available for the upcoming 16 bit microprocessors. UniFLEX™ is the happy marriage of these two previous operating systems. It was designed to run with the "upper class" microcomputer, those which include memory mapping for large amounts of memory. No compatibility restraints were imposed during its design, hopefully eliminating inefficiencies and leaving old design flaws behind.

Many readers will not be familiar with large, multi-user operating systems. In the past, a person's exposure to a computer was typically through a

time-sharing or large computer system. With the advent of the microcomputer, this is no longer true. A large percentage of individuals who deal with computers in their daily work activities have only been exposed to the single-user microcomputer system. With this in mind, the following discussion will point out some of the features needed in a multi-user operating system and how they apply to UniFLEX™.

Multi-user implies that multiple users may make use of a computer's resources simultaneously. This implies that the operating system must be able to perform different operations or tasks simultaneously in order to serve its users' needs. The term multi-tasking describes this ability. Before a user may make use of the system, he must 'login'. This process requires the user to enter a 'user name' and a password. The user name is used by the system to identify the person attempting to access the system, and the password is used to validate the user's authenticity. Most large scale operating systems (including UniFLEX™) will keep a permanent record of the activity of each user. UniFLEX™ keeps a file called 'history' which notes the login and logout times of each user on a terminal by terminal basis. It also keeps information about system boots, date changes, and shut downs in this same file.

As each user attempts to run programs, the operating system must 'schedule' the CPU, allowing it to work on each job. It also must schedule the various I/O devices, such as disk drives, since it is very likely several programs will require disk data all at once. The scheduler is a program internal to UniFLEX™ which determines which task to run, and how long it should be permitted to run. In all cases, the task with the highest priority, which is ready to run, will be selected by the scheduler. A task's priority is determined by many factors but includes such things as memory size, age of task, ratio of I/O to compute time, etc.

Another function of the scheduler is 'swapping'. All large scale operating systems perform swapping but UniFLEX™ is the first to offer it on a micro. 'Swapping out' is the process of copying a task in memory to a secondary memory, such as a disk drive. 'Swapping in' is the reverse, copying the task from the disk back into main memory. The ability to swap tasks allows a system to run many more simultaneous tasks than can be kept in the system's memory. A multi-user or multi-tasking system without it is extremely limited.

As an example, let's consider a system which has 128K bytes of memory, of which, 32K is occupied by the operating system. This leaves 96K free for user programs. Now assume we have a small user environment of four active users on the system. Each user will run a different program, consisting of a text editor (which requires 28K for program and buffers), an assembler (44K), a BASIC interpreter (32K), and a Pascal compiler (40K). A system without swapping would only allow two of the users to start their programs. The other two would get error messages since there is insufficient memory to support their programs. These users would need to keep trying until one of the other user's programs finished. As you can see, this is not very practical.

A system which supports swapping would be able to run all of the programs in the 96K of available memory, even though the total program size is 144K. As above, two users would be able to start running their programs, but when the third and fourth programs are started, again there is not enough main memory. This time, however, the operating system will 'swap' the tasks out to secondary memory (a disk) and not issue the error. The scheduler will see that there are tasks swapped out which are ready to run and proceed to swap the tasks back and forth, effectively sharing the main memory among the various tasks. The swapping operation of course needs to be very efficient to avoid slow down in the system. UniFLEX™, for example, can swap a typical task in about 100 milliseconds (approximately 180K per second transfer rate). This time is using a Winchester type hard disk as a swapping device. The swapping time is 'overlapped' with CPU time which means no time is wasted for a task swap since the processor can be executing another task while a swap is taking place! With this swapping arrangement, there is essentially no memory imposed limit as to the number of active tasks in the system.

Large system operating systems typically offer many user oriented features. Some of these include 'mail', user communications, file protection schemes, and a 'startup' facility. UniFLEX™ offers all of these. The 'mail' program allows one user to send mail to any other user. This mail is simply a message which is tucked away in the receiver's files. Each time a user logs in to the system, he will automatically be informed of any mail received since the last login. This feature is extremely useful in an environment supporting many users. It is also possible to communicate directly with another user who is currently logged in to the system. A bidirectional communications path can be set up between the two users' terminals.

III. The User Interface

After a user 'logs in' to the system, a prompt will be displayed on the terminal, signifying that the system is ready to accept commands. A program, called the 'shell', is responsible for issuing this prompt. The shell is the primary interface between a user and the operating system. It will collect and interpret the commands typed from the terminal, and send the necessary information on to the operating system to perform the requested operation.

All commands in UniFLEX™ have a unique name and the names are somewhat descriptive of the actions performed by each. As an example, typing 'date' will cause the command named 'date' to be executed. This command will display the current date and time on the terminal, just as the name implies. In general, a command line has the following form:

```
command argument1 argument2 ... argumentn
```

The 'command' is the name of the program (file) to be executed, and the 'arguments' are collected and passed on to this program as an array of strings. Notice that the shell does this argument collection, which means individual programs do not have to be concerned with command line parsing. The shell will look several places in the system for the command name specified, including the user's current directory. This allows a user to have 'local' commands with the same names as some system commands.

When a command is executing, the user will usually wait until it finishes, at which time a new prompt will be displayed by the shell. It is possible to 'interrupt' most commands by typing the 'interrupt' character (a control C on most systems). This character will cause premature termination of the command and immediate display of a new prompt. Another similar character is the 'quit' character (a control backslash on most systems) which will do exactly the same thing as the 'interrupt' character, but also create a 'core dump' in the user's directory. A core dump is an exact image of the running program's memory contents at the time the quit character was typed. There are existing system utilities which allow a user to examine this core file as well as the processor registers and stack at time of termination. This feature is obviously a very handy debugging aid.

When a command is executed, it will initially have three files associated with it. These are called the 'standard I/O' files. One 'file' is the user's keyboard (standard input), one is the user's terminal (standard output), and the last one is also the user's terminal and is called the standard error channel. Most commands which perform I/O operations, work with the standard I/O channels.

As an example, the 'list' command will list or display the contents of a file or group of files on the standard output device. Since the standard output is normally the terminal, the file's contents will be displayed on the terminal. The shell has the ability to change the meaning of the standard output to some other file. This process is called output redirection and can be done as follows:

```
list file >outfile
```

This command line would invoke the 'list' command and pass the string 'file' on to 'list' as an argument. The string '>outfile' would not be passed to the command because the symbol '>' has special meaning to the shell. This character tells the shell to redirect the standard output from the terminal, into the file name following. In this example, the output of the list command would go into the file named 'outfile' instead of to the terminal. If this file did not previously exist it would be created, and if it did exist, it would be truncated to zero length before being used. The fact that the shell takes care of this redirection of output means individual commands do not need special code to handle this situation.

Input may also be redirected. As an example, the text editor normally gets its input from the terminal. It is possible to create a file of commands which may be sent to the editor as follows:

```
edit file <script
```

In this case, the file of commands is called 'script' and the input is redirected by the shell as informed by the '<' character. This method of I/O redirection is quite powerful. It should be noted that this convention, as well as most of the other conventions in the UniFLEX™ shell have been closely modeled after the UNIX shell.

The mechanisms involved in the standard I/O scheme can be used to an even greater extent with the implementation of 'filters'. A filter is a program which takes some input data, manipulates this data in some way, and outputs the result. If a program reads the standard input for its data, and outputs its results to the standard output, it can be used in a very powerful way. In particular, the output of one command may be used as input for another command. As an example:

```
sort test-data ↑ reject ↑ spr
```

This command line consists of three commands, 'sort' with the argument 'test-data', 'reject', and 'spr'. The '↑' character is another special character detected by the shell. This separator causes any standard output generated by the command to its left to be sent as standard input to the command on its right. In this example, the sort command will sort the file 'test-data' and send the sorted output to the standard output. Since the shell has set this output to go to the standard input of the next command, 'reject' will operate on this data. Reject is a program which reads the standard input, removes all adjacent duplicate lines, and outputs through the standard output. Again the shell will send this output to the next command, 'spr', which is a system printer spooler. The spooler will take its standard input and print it on the printer. Note that all three commands are essentially run simultaneously. Any data output by the first command is immediately available to the second. This example shows how you can take three totally independent programs and make them work very efficiently together. The mechanism used to 'connect' these filters is called a 'pipe' and is another feature in UniFLEX™ which has been modeled after UNIX. There are many filter type programs in UniFLEX™ and their power should be obvious.

The shell can understand more than one command at a time. As an example:

```
dir; list rugs; date
```

The ';' character is used as a command separator and instructs the shell to continue parsing the command line after the specified command has finished executing. In this example, the commands 'dir', 'list', and 'date' would be executed in a sequential fashion.

It is also possible to have the shell execute multiple commands simultaneously, or in the 'background'. The '&' character used as a command terminator (or separator) will cause the shell to execute the specified command and immediately issue another prompt. As an example:

```
asmb testit >output &
```

will invoke the assembler ('asmb') on the file 'testit' and redirect the output to the file 'output'. Since the command is terminated with a '&', the shell will run the assembly in the background and not wait for it to finish before issuing the prompt. At this time, additional commands may be run, even though the assembly has not completed. The shell will report a 'task identifier' number for all background commands executed. This identifier may be used to terminate the task if desired. Since the '&' may also be used as a command separator, the following is also valid:

```
asmb file1 >out1 & asmb file2 >out2 &
```

This line will cause two assemblies to be run in the background, one on 'file1' and one on 'file2'. These assemblies could have been run sequentially in the background with their output sent to the same file with this command:

```
(asmb file1; asmb file2) >output &
```

The parentheses act like those in expressions, and group parts of the command line which belong together. This same line without the parentheses would have run the assembler on 'file1', sending its output to the terminal. When it finished, a new assembly would be run in the background on 'file2', with it output redirected into the file named 'output'.

As mentioned previously, the shell performs all of the command line parsing, and simply passes the collected arguments on to the executed command as an array of strings. Command line arguments may contain special pattern matching characters recognized by the shell. There are several forms of these matching characters. One is the '*' which matches 'anything'. Another is the '?' which will match any single character. Finally, the construct '[x-y]' will match any character

or range of characters contained in the brackets. Several examples will demonstrate this feature.

```
list text*
asmb source?.a
list *testla-drl
```

The first line will list all files which start with 'text' and have anything following. The second line will assemble the files which start with 'source', have any character next, followed by '.a'. The last example will list all files which end with 'test' followed by one of the characters 'e' through 'd' or the letter 'r'. The shell not only does the matching, but the resulting list of arguments which did match is sorted alphabetically.

Since the shell is no different than any other program, it may also be executed as a command. An application of this is 'command files'. A command file is nothing more than a file containing a list of commands, exactly as they would be entered to the shell. As an example, suppose the two commands 'date' and 'dir' were executed one after the other frequently. A file could be created which contained the following lines:

```
date
dir
```

Assume this file has the name 'dd'. This file can be passed as input to the shell with the following command line:

```
shell <dd
```

Since shell reads the standard input for commands (which is normally the terminal), the input may be redirected to a file. In this example, the shell will read the file, and execute each command, date and dir. The shell will then terminate. This example is not very useful, but suggests how very complex command files may be constructed and executed. It is actually possible to directly execute a command file without having to type 'shell <', but this method will not be described here.

The shell has many more features. Since it is the primary interface between the user and the system, it is very important that it be powerful and easy to use. The UniFLEX™ shell appears to be both, and will undoubtedly gain additional features in the future.

To be continued

Flex User Notes

BY: RONALD W. ANDERSON
3540 STRUBRIDGE COURT
ANN ARBOR, MI 48105

JOB CONTROL PROGRAM

A few days ago, I received a copy of JCP from Frank Hogg. Since he didn't ask for any kind words here, I will give him some. JCP was reviewed in the July issue, and I won't repeat a review here. JCP kind of grows on one. At first glance, it is a substitute EXEC utility. Exec has one flaw. It is not possible to pass it any parameters. Suppose you want to make a JCP file that will let you recompile your Pascal program a number of times without going through all the delete and compile commands from the keyboard. You could use EXEC, but you would have to write a new command file for every program you want to recompile. That is because the file must include the name of the program to be recompiled. The sequence of commands for Lucidata Pascal and a program named PRIMES would be as follows manually:

```
+++DELETE PRIMES.BAK
(RESPOND Y Y TO THE PROMPTS)
+++EDIT PRIMES
+++RUN PASCAL PRIMES
(RESPOND TO STACK SIZE PROMPT 1800)
+++RUNLGO,PRIMES
```

Using JCP you would set up a command file called RECOMP or some such.

```
ONERROR CONT
DELETE $1.BAK
ONERROR BREAK
BREAKN
EDIT $1
RUN PASCAL $1
RUNLGO $1
END
```

The command line would be RECOMP'PRIMES. The ' is the delimiter for parameters passed to the command file. This works exactly like a Macroassembler. If you have ever seen one you will immediately see the similiarity. The string "PRIMES" is substituted wherever $1 appears in the command file. Now to recompile TEST, you only change the command line. The command files you may write become new utilities that may use a number of other programs to run.

One of the nice things about JCP is that a complete and well commented source listing is supplied, as well as many examples of JCP programs. The disappointment to me was that JCP is not available for FLEX9. I tried to modify the source, but Peter Murray, JCP's auth had to do some overlay modifications to FLEX2, (which are undone on exiting JCP), and I have not been successful in translating all of them to work with FLEX9.

RESPONSES

Responses to my column have begun to appear in my mail. Some of you have made suggestions for topics for discussion here, and others have expressed some pleasure with what we are doing each month. My article on "The Perfect Compiler" brought mail from Douglas Beck who has Motorola's MPL running, and thinks it is "the best of a doubtful lot". It lacks floating point capabilities, but has Bit, Byte, Word, and Char types of variables which must be declared prior to use as in SPL/M or Pascal. The compiler allows imbedded assembly language easily, as the output of the compiler is a source file for a macro assembler. Doug points out that this means a multi step compile operation but indicates that it is not much of a problem. I have found that such compile operations may be made simpler by using an EXEC file, and could be made ultra simple with JCP (see above). The PL languages are similar to Pascal in that Do-While loops are implemented, as well as IF-THEN-ELSE. For the applications I have in mind, MPL would be very good if it supported floating point arithmetic. MPL is a product of Motorola Microsystems. It is supplied for their Exorciser system, compatible with MDOS. Doug didn't indicate whether he or anyone else has converted it to run in FLEX. This is definitely not a hobbyist compiler. The price tag from Motorola is around $1500!

I asked Doug to write my prime numbers program in it and report on the running time to find the primes between 1 and 1000. I have used this as a test on several compilers, and I will rep t on the time and memory usage for several of them here.

I received a letter from Robert Peirce voting for the Software Dynamics BASIC compiler. I have just purchased their compiler and I am impressed with it even though it is BASIC, and it has a very large run time package. Also, it is not very efficient in terms of code generated, ie. memory usage. SD BASIC, is however, very capable, and would be a good choice for a compiler for business software that is to be run on a microcomputer system (what I call a bench system as distinguished from an industrial application system in which the program is fixed in EPROM for the life of the machine that

it controls or monitors). Bob indicated that SD has a new version 1.4 BASIC that is not available (yet) for FLEX, but runs in his MSI system. This version allows labels as well as variable names, and produces very well documented code. Bob sent some examples of his programming that indicate that much can be done in a structured way with SD BASIC.

Interestingly, I have not heard from anyone who shares my enthusiasm for Lucidata Pascal. I have written about 30 pages of programs for machines using this Pascal, and find it very easy to write and debug programs compared to doing the same thing in Assembler. The rule that a line of code takes the same amount of time to debug, whether in Assembler or a higher level language seems to be holding. The listings are about 1/8 the size of the equivalent program in Assembler, and the debug time is about the same ratio. My only wish is that Lucidata would have produced a little more complete implementation, particularly with regard to the WITH statement that allows reference to the parts of a RECORD data structure without repeating the whole name endlessly.

I have it from Dan Vanada of TSC that their C and Pascal compilers are coming along in a couple of months, and will both be integer only initially. They will later have a floating point version of each. These will be "native code" compilers, ie. no intermediate code. That means that there will be no license problem with a run-time package (says Dan Vanade). That means that I can write utilities in their Pascal and sell them without requiring you to buy their run time package, or having to pay them a huge fee for a license to sell (or give away) a run time package so you can run my utilities, games, etc.

TYPING ANYONE?

I've started a rather interesting project. I have begun to write a series of typing lessons in TSC BASIC. The set is designed to teach a non-typist (or two finger variety) to type by the "touch" system on his terminal. If you haven't guessed, I type rather rapidly on my terminal, having had a course in typing in High School. In college, I found that my handwriting was so bad, that I brought my portable typewriter to school to do the engineering lab reports. I type fast, but I make a lot of mistakes, so the terminal is really great for me. I have added the 'CT L'H to my touch typing. I know without looking that I have made an error, and can back up and correct it automatically. When I am done, I will probably publish the first lesson here, and offer the whole set on a disk. The lessons consist of a great deal of drill, and a reporting of all errors. Each line must be typed correctly before another will be presented. I found that I improved my own typing significantly just trying out the lessons. I need a great deal of drill on punctuation and numbers, (the top row of keys on the terminal).

ONE FILE RATHER THAN TWO

Shortly before this column started appearing in '68' Micro Journal I sent some back issues of my newsletters to Francis Van orn. He wrote responding to my Modem programs (I had used a M.CMD and a MODEM.SYS patterned after P.CMD and PRINT.SYS). Francis pointed out that the function of loading and initializing the print routine doesn't require the use of two separate files. Sometimes the obvious isn't. Since Francis' program was in Miniflex, I modified it and appended my print routine that seems to work well with my Pa r Tiger. The listing is supplied here. You may use your own PRINT.SYS rather than mine. The program works just as well and only takes one sector on the disk. Of course a few equate changes would make it work for FLEX2 just as well. "Great", I said, and then I loaded BASIC and tried printing by 'OPEN " .PRINT" AS 0' and then using some PRINT #0, statements. I got an error on the open statement. Apparently BASIC doesn't fool with P.CMD but loads PRINT.SYS directly. TSC has been kind enough to scramble all text strings in BASIC so that they don't appear in an ASCII dump. Therefore, the string can't be found and changed. Oh well, PRINT.SYS was put back on the disk and BASIC was again happy. A few nights later, it occurred to me that perhaps OPEN "O.PRINT" opens PRINT.SYS, so I tried OPEN "O.P.CMD" AS 0 in a BASIC program (remember that my new P.CMD also loads what used to be PRINT.SYS), and the result was complete success. Goodby PRINT.SYS, and hello extra sector on the system disk. Of course the modem programs I published here a couple of months ago, could be rewritten similarly to eliminate a file and a sector.

I can only think of one reason TSC would have written FLEX to use the two separate files. P.CMD is larger, and is standard regardless of the type of printer you have. Your PRINT.SYS is the variable part of the program, and by making it a separate file, they have simplified writing a s cial printer driver for various printers.

```
U (UTILITIES) SHORT COMMANDS          5-10-80  TSC ASSEMBLER  PAGE    1

  5                       *
  6                       *
  7                       * THIS IS A MINI MONITOR. IT CONTAINS SEVERAL VERY SHORT UTILITIES
  8                       * THAT ARE ACCESSED BY A KEY LETTER AND ANY ARGUMENTS THAT THEY
  9                       * MAY NEED. PASSED ON THE COMMAND LINE.
 10                       *
 11                       * KEY LETTERS ARE:
 12                       *
 13                       * F - FILL, FOLLOWED BY BYTE,STARTADDRESS,ENDADDRESS
 14                       *     U;F,00,0100,01FF
 15                       *
 16                       * H - OUTPUT HEX DOUBLE BYTE, FIRST BYTE MAY BE NULL
 17                       *     U;H;0D0A
 18                       *
 19                       * L - LOAD, FOLLOWED BY FILENAME, OFFSET
 20                       *     U;L,TEST.TXT,1,1000
 21                       *
 22                       * N - NO PAUSE  NO ARGUMENTS REQUIRED
 23                       *
 24                       * P - PAGE  NO ARGUMENTS, OUTPUTS FORMFEED CHARACTER
 25                       *     P;U;P
 26                       *
 27                       * R - RESTORE PAUSE
 28                       *
 29                       * S - SKIP NUMBER OF LINES FOLLOWING (DECIMAL)
 30                       *     P;U;S;6
 31                       *
 32                       *
 33                       * THE IDEA IS TO COMBINE SEVERAL SIMPLE COMMANDS IN ONE FILE SECTOR
 34                       * TO SAVE SECTOR AND DIRECTORY SPACE ON A SYSTEM DISK.
 35                       *
 36
 37
 38                       *
 39                       *  FLEX2 EQUATES LIBRARY
 40                       *
 41  D000                 PORT0   EQU   $D000
 42  D004                 PORT1   EQU   $D004
 43  D008                 PORT2   EQU   $D008
 44  D00C                 PORT3   EQU   $D00C
 45  D010                 PORT4   EQU   $D010
 46  D014                 PORT5   EQU   $D014
 47  D014                 DRVREG  EQU   $D014     DRIVE SELECT REGISTER
 48  D018                 PORT6   EQU   $D018
 49  D018                 CNTREG  EQU   $D018     DISK CONTROL REGISTER
 50  D019                 TRKREG  EQU   $D019     DISK TRACK REGISTER
 51  D01A                 SECREG  EQU   $D01A     DISK SECTOR REGISTER
 52  D01B                 DATREG  EQU   $D01B     DISK DATA REGISTER
 53  D01C                 PORT7   EQU   $D01C
 54
 55  2400                 LOADER  EQU   $2400     FLEX DISK LOADER ADDRESS
 56
 57  A00C                 PORECH  EQU   $A00C
 58
 59  AB40                 FCB     EQU   $AB40     FILE CONTROL BLOCK
 60
 61                       * TTYSET CHARACTER PARAMETERS
 62                       * "D=$00" MEANS DEFAULT VALUE IS HEX 00
 63                       * 'L' MEANS CONTROL CHARACTER L
 64                       * "(NUL) MEANS ASCII CONTROL CHARACTER IN THIS CASE A NULL
 65                       *
 66  AC00                 BSPCHR  EQU   $AC00     BACK SPACE D=$08 ^H (BS)
 67  AC01                 DELCHR  EQU   $AC01     DELETE CHAR D=$18 ^X (CAN)
 68  AC02                 EOLCHR  EQU   $AC02     END OF LINE CHAR D=$3A : (COLON)
 69  AC03                 DEPTH   EQU   $AC03     DEPTH COUNT D=0
 70  AC04                 WIDTH   EQU   $AC04     WIDTH COUNT D=0
 71  AC05                 NULLS   EQU   $AC05     NULL COUNT D=4
 72  AC06                 TABCHR  EQU   $AC06     TAB CHAR D=0
 73  AC07                 BSPECH  EQU   $AC07     BACK SPACE ECHO CHAR D=0
 74  AC08                 EJECT   EQU   $AC08     EJECT COUNT D=0
 75  AC09                 PAUSE   EQU   $AC09     PAUSE CONTROL D=$FF $00=NO PAUSE
 76  AC0A                 ESCAPE  EQU   $AC0A     ESCAPE CHAR D=$1B   (ESC)
 77  AC0B                 SYSDRV  EQU   $AC0B     SYSTEM DRIVE NUMBER D=0
 78  AC0C                 WORKDV  EQU   $AC0C     WORKING DRIVE NUMBER D=0
 79  AC0D                 SYSTEM  EQU   $AB00     RESERVED FOR SYSTEM
 80  AC0E                 DATE    EQU   $AC0E     DATE REGISTER
 81  AC0E                 MONTH   EQU   $AC0E     MONTH BYT
 82  AC0F                 DAY     EQU   $AC0F     DAY  YTE
 83  AC10                 YEAR    EQU   $AC10     YEAR BYTE
 84  AC11                 LSTTRM  EQU   $AC11     LAST TERMINATOR
 85  AC12                 USRADR  EQU   $AC12     USER COMMAND TABLE ADDRESS
 86  AC14                 BUFPNT  EQU   $AC14     LINE BUFFER POINTER
 87  AC16                 ESCREG  EQU   $AC16     ESCAPE RETURN REGISTER
 88  AC18                 CURCHR  EQU   $AC18     CURRENT CHARACTER
 89  AC19                 PRECHR  EQU   $AC19     PREVIOUS CHARACTER
 90  AC1A                 CURLIN  EQU   $AC1A     CURRENT LINE NUMBER
 91  AC1B                 LDOFST  EQU   $AC1B     LOADER ADDRESS OFFSET
 92  AC1D                 TRNSFG  EQU   $AC1D     TRANSFER FLAG
 93  AC1E                 TRNSAD  EQU   $AC1E     TRANSFER ADDRESS
 94  AC20                 ERRNUM  EQU   $AC20     ERROR NUMBER FROM FMS
 95
 96  A000                 BUFFER  EQU   $A080
 97  A0FF                 BUFEND  EQU   $A0FF
 98
 99  A000                 COLDS   EQU   $A000
100  A003                 WARMS   EQU   $A003
101  A006                 RENTER  EQU   $A006
102  A009                 INCH    EQU   $A009
```

```
103  AD0C           INCH2   EQU   $AD0C
104  AD0F           OUTCH   EQU   $AD0F
105  AD12           OUTCH2  EQU   $AD12
106  AD15           GETCHR  EQU   $AD15
107  AD18           PUTCHR  EQU   $AD18
108  AD1B           INBUFF  EQU   $AD1B
109  AD1E           PSTRNG  EQU   $AD1E
110  AD21           CLASS   EQU   $AD21
111  AD24           PCRLF   EQU   $AD24
112  AD27           NXTCH   EQU   $AD27
113  AD2A           RSTRIO  EQU   $AD2A
114  AD2D           GETFIL  EQU   $AD2D
115  AD30           LOAD    EQU   $AD30
116  AD33           SETEXT  EQU   $AD33
117  AD36           ADDBX   EQU   $AD36
118  AD39           OUTDEC  EQU   $AD39
119  AD3C           OUTHEX  EQU   $AD3C
120  AD3F           RPTERR  EQU   $AD3F
121  AD42           GETHEX  EQU   $AD42
122  AD45           OUTADR  EQU   $AD45
123  AD48           INDEC   EQU   $AD48
124  AD4B           DOCMND  EQU   $AD4B


126  B401           FMS LS  EQU   $B403
127  B406           FMS     EQU   $B406
128
129  BE80           READ    EQU   $BE80    READ  CTOR
130  BE83           WRITE   EQU   $BE83    WRITE  ECTOR
131  BE86           VERIFY  EQU   $BE86    VERIFY SECTOR
132  BE89           RESTOR  EQU   $BE89    RESTORE TO TRACK ZERO
133  BE8C           DRSLCT  EQU   $BE8C    DRIVE SELECT
134  BE8F           DREADY  EQU   $BE8F    DRIVE READY
135  BE92           QREADY  EQU   $BE92    QUICK DRIVE READY
137                 *
138  A100                   ORG   $A100
139
140  A100 20 07     START   BRA   BEGIN
141  A102 01        VN      FCB   1         VERSION NUMBER (FLEX CONVENTION)
142  A103           BEGADD  RMB   2         POINTER FOR BEGINNING ADDRESS FOR FILL, MOVE
143  A105           ENDADD  RMB   2         POINTER FOR ENDING ADDRESS
144  A107           BYTE1   RMB   1         FIR T BYTE OF A DOUBLE BYTE POINTER
145  A108           BYTE2   RMB   1         SECOND BYTE
146                 *
147  A109 BD AD 27  BEGIN   JSR   NXTCH     COMMAND CHARACTER
148  A10C 36                PSH A           SAVE CHARACTER
149  A10D BD AD 27          JSR   NXTCH     GET ANOTHER
150  A110 32                PUL A           RECOVER THE COMMAND LETTER
151  A111 CE A1 C6          LDX   #JTAB     POINT AT THE JUMP TABLE
152  A114 A1 00     PROC1   CMP A 0,X       COMPARE COMMAND LETTER WITH TABLE ENTRY
153  A116 27 0A             BEQ   FOUND1    IF SAME, COMMAND IS FOUND
154  A118 08                INX
155  A119 08                INX
156  A11A 08                INX             TRY NEXT ONE
157  A11B BC A1 DE          CPX   #ENDJAB   TRIED ALL ENTRIES  ET?
158  A11E 27 06             BEQ   #ERR      NOT FOUND
159  A120 20 F2             BRA   PROC1     IF MORE TO TRY, GO AROUND AGAIN
160  A122 EE 01     FOUND1  LDX   1,X       POINT TO THE START ADDRES  FOR COMMAND
161  A124 6E 00             JMP   0,X       JUMP THERE
162                 *
163                 * ERROR ROUTINES
164                 *
165  A126 CE A1 DE  #ERR    LDX   #NFSTR    NOT FOUND
166  A129 BD AD 1E          JSR   PSTRNG    PRINT IT TO TERMINAL
167  A12C 7E AD 03          JMP   WARMS     EXIT TO FLEX
168                 *
169                 * FILL ROUTINE
170                 *
171  A12F BD AD 42  FILL    JSR   GETHEX    GET THE BYTE TO FILL WITH
172  A132 FF A1 07          STX   BYTE1     GETHEX GETS VALUE IN X, DOUBLE BYTE
173  A135 BD AD 42          JSR   GETHEX    GET STARTING ADDRESS
174  A138 FF A1 03          STX   BEGADD
175  A13B BD AD 42          JSR   GETHEX    GET ENDING ADDRESS
176  A13E FF A1 05          STX   ENDADD
177  A141 B6 A1 08          LDA A BYTE2     ET FILL BYTE
178  A144 FE A1 03          LDX   BEGADD
179  A147 A7 00     FILL1   STA A 0,X
180  A149 BC A1 05          CPX   ENDADD
181  A14C 27 03             BEQ   DONFIL
182  A14E 08                INX
183  A14F 20 F6             BRA   FILL1     LOOP UNTIL END A BD 96 IS REACHED
184  A151 7E AD 03  DONFIL  JMP   WARMS     EXIT TO FLEX
185                 *
186                 * HEX OUTPUT ROUTINE
187                 *
188  A154 BD AD 42  HOUT    JSR   GETHEX    GET HEX DOUBLE BYTE TO OU PUT
189  A157 FF A1 07          STX   BYTE1
190  A15A B6 A1 07          LDA A BYTE1     GET FIRST BYTE
191  A15D BD AD 18          JSR   PUTCHR    OUTPUT IT
192  A160 B6 A1 08          LDA A BYTE2     GET SECOND BYTE
193  A163 BD AD 18          JSR   PUTCHR    OUTPUT IT
194  A166 7E AD 03          JMP   WARMS
195                 *
196                 * JUMP COMMAND
197                 *
198  A169 BD AD 42  JUMP    JSR   GETHEX    GET ADDRESS TO WHICH TO JUMP
199  A16C 6E 00             JMP   0,X       DO IT
200                 *
201                 * LOAD FILE WITH OFFSET
202                 *
203  A16E CE AB 40  LOADO   LDX   #FCB      POINT AT FILE CONTROL BLOCK
204  A171 BD AD 2D          JSR   GETFIL    GET FILE SPECS
205  A174 BD AD 42          JSR   GETHEX    GET LOAD OFFSET
206  A177 FF AC 1B          STX   LDOFST    LOCATION IN FLEX
207  A17A CE AB 40          LDX   #FCB
208  A17D C6 01             LDA B #1        OPEN FOR READ CODE
209  A17F E7 00             STA B 0,X
210  A181 BD B4 06          JSR   FMS       OPEN THE FILE FOR READ
211  A184 C6 FF             LDA B #$FF
212  A186 E7 3B             STA B 59,X      TURN OFF SPACE SUPPRESSION FLAG
213  A188 BD AB 30          JSR   LOAD      LOAD THE FILE
214  A18B 7E AD 03          JMP   WARMS
215                 *
216                 * TURN PAUSE OFF
217                 *
218  A18E 7F AC 09  NOPAUS  CLR   PAUSE     LOCATION IN FLEX
219  A191 7E AD 03          JMP   WARMS
220                 *
221                 * PAGE OUTPUT
222                 *
223  A194 86 0C     PAGE    LDA A #$0C      PAGE CONTROL CHARACTER ON MOST PRINTERS
224  A196 BD AD 18          JSR   PUTCHR
225  A199 7E A0 03          JMP   WARMS
226                 *
227                 * RESTORE PAUSE
228                 *
229  A19C 86 FF     RESTR   LDA A #$FF
230  A19E B7 AC 09          STA A PAUSE
231  A1A1 7E AD 03          JMP   WARMS
232                 *
233                 * SKIP N LINES
236                 * ARGUMENT AFTER $ IS ASSUMED A TWO DIGIT DECIMAL NUMBER
235  A1A4 BD AD 42  SKIP    JSR   GETHEX    GET NUMBER TO SKIP
236  A1A7 FF A1 07          STX   BYTE1
237  A1AA F6 A1 08          LDA B BYTE2     GET LOW ORDER BYTE
238  A1AD 5D        SKIP1   TST B
239  A1AE 27 13             BEQ   DONSKP    DONE IF ZERO
240  A1B0 86 0A             LDA A #$0A      LINEFEED
241  A1B2 BD AD 18          JSR   PUTCHR
242  A1B5 5A                DEC B
243  A1B6 27 03             BEQ   DONSKP
244  A1B8 17                TBA
245  A1B9 B4 0F             AND A #$0F      MASK OFF HIGHE  DIGIT
246  A1BB B1 0F             CMP A #$0F      IF EQUAL, CARRY AND DECIMAL ADJUST
247  A1BD 26 02             BNE   SKIP2     NO CARRY OR DECIMAL CORRECTION
248  A1BF C4 F9             AND B #$F9      DECIMAL CORRECTION
249  A1C1 20 EA     SKIP2   BRA   SKIP1
250  A1C3 7E AD 03  DONSKP  JMP   WARMS
251                 *
252                 * JUMP TABLE HERE
253                 *
254  A1CA           JTAB    EQU   *
255  A1 6 46                FCB   'F
256  A1C7 A1 2F             FDB   FILL
257  A1C9 48                FCB   'H
258  A1CA A1 54             FDB   HOUT
259  A1CC 4A                FCB   'J
260  A1CD A1 69             FDB   JUMP
261  A1CF 4C                FCB   'L
262  A1D0 A1 6E             FDB   LOADO
263  A1D2 4E                FCB   'N
264  A1D3 A1 8E             FDB   NOPAUS
265  A1D5 50                FCB   'P
266  A1D6 A1 94             FDB   PAGE
267  A1D8 52                FCB   'R
268  A1D9 A1 9C             FDB   RESTR
269  A1DB 53                FCB   'S
270  A1DC A1 A4             FDB   SKIP
271  A1DE           ENDTAB  EQU   *
272                 *
273                 * STRING FOR ERROR MESSAGE
274                 *
275  A1DE 42        NFSTR   FCC   /BAD COMMAND/
2 6  A1E9 04                FCB   4
277                 END     ST  E

NO ERRORS DETECTED
```

WINNERS
Giant Software Contest

A very difficult task is now finished!! The final decisions are in on the GIANT SOFTWARE CONTEST and listed below, for those awarded 1st place as well as the additional vendor donated prizes. Space will not permit the listing of all winners.

Winners in all places should have received notice by now of their prize. Those items donated by various advertisers and other dedicated 68XX vendors and users will be forwarded from the donor. We have informed each donor of the winners name and address, for all prizes listed in a previous issue of 68 Micro Journal (Nov/Dec '79). Those winning subscriptions have been notified by the staff of 68 Micro Journal and their subscription labels should reflect the change within the next issue or so.

GRAND PRIZE GIMIX 6809 System:
Brian F. Bailey WB4MMP

GRAND PRIZE SWTPC 69K System:
Curtiss C. Barrett

GRAND PRIZE SSB $500.00 Gift Certificate:
Dan Lipnisky
Dan Johnson

GRAND PRIZE TSC ($250.00) Software:
John Jordon

GRAND PRIZE DIGITAL RESEARCH: Computers:
  16K Static Memory Board
Mickey Ferguson

GRAND PRIZE SOFTWARE DYNAMICS:
  BASIC Compiler PLUS Assembler
R. B. Pierce

GRAND PRIZE COMPUTERWARE $200.00 Gift Certificate:
Jeffery Brownstein

GRAND PRIZE MICROWORKS $179.95 Gift Certificate:
James Hughes

GRAND PRIZE JPC Hardware and Software:
  *AD-16
Brent Ermlick
  *Ck-7
Roger Schaefer
  *TC-3 w/CFM-3 software
David Dawson

GRAND PRIZE SSI $100.00 Software:
N. E. Barnes

GRAND PRIZE MICROWARE ABASIC Compiler, Source Gen,
  Interpreter and Lisp:
Bryan Loofbourrow
 *RT 68 ROM's and 6800 Chess:
John Gildewell
Joe Dubner
James Caraway
David Dawson
Marc Provencher
Leo Garrett

GRAND PRIZE LUCIDATA PASCAL:
Mark Sproul
L. L. Stevens
James McVay
Joe Turner Jr.
John P. Tucker

GRAND PRIZE HEMENWAY ASSOCIATES, INC. Books:
  *SP/68 Operating System
Wolfgang Meister
Karl Vollbrecht
James Shivia
  *Macro Linking Cross Assembler
Allen Clark
John Tarvin
Rudolf Reuter
  *STRUBAL+
William Pfeifer
Frank Hoffman
Ronald Schmars

GRAND PRIZE STAR-KITS Gift Certificate:
Paul Phelps

GRAND PRIZE HHH Enterprises and Springbok:
  *SPIRIT and SO-1
Dan Lipnisky

GRAND PRIZE CER-COMP MiniDisk+ (EPROM) and software:
Bob Jacobs

The following are TOP WINNERS of Life subscriptions from 68 MICRO JOURNAL ($250.00 each).

Curtis Barrett, N. E. Barnes, John Jordon, Dan Lipnisky, Carl Kreider, Dan Johnson, Steve Tabler, R. B. Pierce, Wilson Federici, Frank Barney, Paul Phelps, Brent Ermlick, David Eagle, Alfred Dodd, James Dodgen Jr., James Hughes.

In addition there are over 300 other winners of 68 MICRO JOURNAL subscription extensions of from one to three years.

CONGRATULATIONS ALL!!!!!!!!!!!!!!!!

My sincere appreciation to all those who donated the wealth of prizes to the contest. Also I want to give a special thanks to all those who gave of their time to judge the entries.

As stated a month ago, I know that had the judges been different, the results would have also been different in many cases. This was an undertaking of far greater magnitude than I had ever envisioned. At no time did we ever get everyone (judges) to agree upon the order of awards. However, the decisions were made on a majority basis. And I want to thank each and every one who participated!

I am not sure that we will ever do it again. Maybe we will, and if we do I will be better prepared to get it finished with a bit more promptness. We all sure learned a lot! Our overseas entries had a lot of disks bombed, this required months to get straightened out. Even by airmail(?) it required over 10 weeks to get a reply back in a few instances. In most all cases it took far longer than we anticipated. If babies were delivered by mail, many would be eligible for social security before their first birthday.

We will commence in the near future to publish the entries. Some are large and will have to be broken into several issues. It will take some time to run them all, but it will have been worth waiting for.

Please note the manufacturers and vendors (advertisers) who donated. It would be good if the next time you order something from them to let THEM know that you appreciate their support by donating. As for me, in all my 50+ years, I have never had the privilege of knowing a better group of folks than YOU - 68XX users.

DMW ——

Save that SWTPC MPA CPU

Garry O Caudell
3125 Robin Lynn
Ashland, Ky 41101

I know they said it couldn't be done, but would you like to run FLEX-2 on your old MPA CPU board? Would you like to convert your MPA2 CPU board to 6809 and run FLEX-9 without modification? Want to add a whole bunch of esoteric switches to your computer? If you answer yes to any of these question read on!

After I had converted my MPA2 board to 6809 I ran into a slight problem. (more on the conversion later) I was able to convert it back by changing the PERCOM adapter board for a 6800 chip, replacing the 6875, throwing switches on the CPU and a switch on the mother board.

What I actually was doing was putting my old MPA board in and throwing the motherboard switch when I

wanted to run 6800 programs (which was quite often since most of my programs were in 6800).

About then a friend of mine, who only had a MPA CPU, was moaning about not being able to have random files and guessing that his only recourse was to go to the 6809 board as SWTPC was dropping support of the 6800.

Since I hadn't been going to the trouble to convert my MPA2 board back to 6800 and my friend couldn't run FLEX-2 I took a look to see what had to be done to run FLEX-2 on a MPA CPU board.

As it turned out it was not all that hard, so now I can run MINI-FLEX or FLEX-2 on my MPA board and FLEX-9 on my (converted) MPA2 board.

The problem with using FLEX-2 is that it is located from $A080 to $BFFF and the monitor ram uses $A000 to $A07F for a scratch pad. This is further complicated by the on board RAM being only partially decoded and in fact ties up 8 K of memory addresses. On the MPA2 board it is possible to turn off the on board RAM and put some memory at $A000 and then let FLEX-2 start at $A080.

Looking at the schmatic, IC's 16,13 and 12 are used to decode address lines and uses this information to keep the data lines from going off the board. Also they generate the on board chip select for the ROM and RAM. IC-16 has one gate (pins 9,10,12,13) tied to address lines A-15 and A-13. Its output is is used along with some other signals to gate the data lines (IC-8 and IC-9) and to the enables on ROM and RAM. This is telling the CPU tha anything where A15 and A13 are high stay on the CPU board. This means the anything with address $F000,$E000, $B000 and $A000 stays on the CPU. Well this works out ok. The ROM is at $E000 to $E3FF but has to also pretend To be at $FFF0 to $FFFF for the restart vectors. and the scratch pad is at $A000. So the scratch pad ties up one 8 K block and the monitor ties up another. Off board high memory is allowed at $C000-$DFFF.

To fix all this up we need to add A-14 to the logic. To make this easy, IC-16 is a quad nand gate with only two inputs. The inputs are used in pairs, so we can cut loose one of the unused inputs. This will mean that only $E000-$F000 will stay on the CPU board. To do this cut the trace between pins 10 and 13 on IC-16. A-14 can be located just to the right of the ground bus that runs up the middle of the board. It's the third feed-though hole form the bottom. Connect pin 13 to A14. I put mine through a switch with IC-16 pin 13 on the center arm, IC-16 pin 10 on one end and A-14 on the other end. This allows you to restore the original circuit when you want to. Guess I just like switches.

Address a memory board for $A000 to $BFFF and turn on the system. You should get the usual "$" prompt. Go to $A100 with the M command and put some pattern in memory that you will recognize, then go look at $A000. You should not see the same pattern. Before the modification the pattern would have been repeated. Boot up Flex-2 and have fun. Mini-Flex will boot normal, without throwing the switch, just dont let any of your friends know that you are using an 8 K board for only 128 bytes.

Like most other 6800 hobbyists, I was interested in the new 6809 microprocessor. I had not got involved until the article by Dr. Pentecost came along in 68 micro-journal 2,2 came out.

I followed his article exactly had a couple of minor problems First, there seemed to be an address error in the table ($C626 shoulb be $C62C) for patching the NEWDISK.CMD. Secondly, it was not clear that the new NEWDISK.CMD should be saved using the SAVE.LOW command.

Using the SAVE.CMD resulted in my saving the save command. Then, when I attempted to delete this garbage I succeded in erasing the original NEWDISK.CMD as my copy of Flex9 was unable to tell the difference between NEWDISK.CMD and NEWDISK2.CMD. If you go his route call your new NEWDISK.CMD something different like NEW80.CMD. This is not to nit-pick or to find fault with Dr. Pentecost's fine article, just trying to keep someone else from making the same dumb mistakes I did.

After gettin my 09 system up and running I decided that I would look into a way to move my I/O to the $E000 address used in the SWTPC version of the 09. I have felt that the main advantage of the 6800 systems was poor old dumb MIKBUG. Mikbug, poor as it is, prevented us from running off in all directions as others have done. We all have our monitors in the same place, our I/O in the same place and our RAM in the same place. So, even though MIKBUG messes up 32 K of memory we can all talk to each other. MIKBUG was cheap and available when we needed it. For this reason I felt that if we created a sub group of 6809's using FLEX-9 with $8000 I/O we would eventally be sorry. The solution I came up with seems very reasonable. Cut one board trace (MP-A2) and either install a jumper or put it through a single-pole-double-throw switch if you want to make it easy to change back to a 6800 system.

First modify the motherboard for $E000 I/O To do this see the article by Alexander in 68-Micro 2-4. Or see the instuctions that SWTPC puts out with the new 6809 CPU board.

Next, let's look at the CPU. The address lines are all right. The CPU board has some fairly elobarate circuitry to keep the data lines from being enabaled to the motherboard whenever something is supposed to be on the CPU board. IC's 13,15,16,17 and the dip switches are all involved in this. The switches and these chips all act to enable or disable IC's 19 and 20.

There is no provision in all of this for anything in the $E000 to $F000 range to come off the CPU. This is MIKBUG/SWAT BUG territory. We want our I/O to be at $E000 and our monitor to be at $F800. Since each of the 2716 socets have to be enabled I cut the trace from IC-15 pin 12 to IC-16 pin 2. I then jumpered IC15 pin 12 to IC-13 pin 9. This allows everything to go off the CPU except $F800-$FFFF This is the correct address for the SBUG-E monitor. I later changed this to a SPDT switch to to make this connection or restore the original so I could go back to the 6800.

I have my dip switches set 4K-8K,8K,HI-PROM on and all others off.

To date I have had no trouble with the PERCOM 6809 adapter board. I first brought my system up with PSYMON but was soon wanting to use my disks. I then brought up SWBUG modified for $8000 Before tackling the $E000 modification. This allowed me to check myself at each step. The PSYMON step is not necessary as there are no further modifications to do to SWBUG at $8000. The only thing I have lost is the Dynamic Address Translator. I dont know of anything that this deprives me of, except I will have to be careful to address my memory in sequence. Small loss for the difference in cost.

I found that if the 2 for 79 cent RADIO-SHACK slide switches (part # 275-407) are cleaned and tinned they can be soldered to any convenient plated area on the printed circuit board. If you dont mark them with a felt pen, you will be the only one to know what they are.

Diagrams Continued on Page 21...

TRS-80 to S50 Bus

CLOAD 1

A TRS-80 Level I Tape Loader for the 6800 Micro

Terry Mayhugh
11632 Midhurst Dr.
Concord, Tn. 37922

As a fellow 6800 enthusiast once remarked "one can't help but notice the large amount of software available for the TRS-80." I felt the same way especially after acquiring Percom's Electric Crayon which brought multi-mode graphics capability to my SWTP 6800/2 computer. With 10 modes of color graphics to choose from it is not difficult to find a level of resolution compatible with the Apple, PET, or TRS-80 micros. So, I wrote CLOAD 1 as a first step toward taking advantage of some of the TRS-80 BASIC software which is so widely distributed. Using this program in conjunction with the simple recorder interface described later, level I BASIC tapes may be loaded into a 6800 microcomputer. Although my particular system runs under Smoke Signal's DOS 68.51 with SWTBUG (and with TSC's text editor) only minor program changes should be required for other operating systems. CLOAD 1 was written to load the TRS-80 program into the TSC text editor buffer area. The excellent global editing commands of this editor make any necessary program modifications a snap. After editing, the modified TRS-80 program is saved to disk as a text file for later use by a BASIC interpreter. TSC text files are compatible with nearly all 6800 BASIC interpreters.

Since the TRS-80 tape format has been examined in detail in several past articles, I will only briefly describe the level I format for the benefit of those 6800 users who weren't paying attention previously. Basically, the format is similar to that of a floppy with sync pulses occuring every 2 msec and data bits occuring midway between. A level I BASIC tape consists of 128 zero bits for a leader, a sync byte (hex A5), a two byte beginning-of-load address, a two byte end-of-load address, the program data (8 bit ASCII, no parity), and finally a 2's complement checksum. Figure 1a shows the tape pulse timing relationship.

The simple interface shown in Figure 2 was constructed to convert the recorder pulses into cleanly shaped pulses compatible with an MPLA parallel interface card. The B side MSB of an MPLA card located in slot #7 is used as the data input to the program. The full wave adaptive threshold technique of Goetz and Miller in an earlier 73 article was borrowed with excellent results. In fact, using this interface you will probably be able to read TRS-80 tapes more easily than many TRS-80 owners! Figure 1b above the expected output from the interface. A 150 μsec TTL compatible pulse is generated for each sync or data pulse read from the tape. A 600 μsec lock out period is generated as in the Goetz and Miller E-Z loader during which output due to tape noise, etc. is inhibited. Actually, this feature is not required by CLOAD 1 since adaptively timed software gates are opened only at appropriate times to look for data bits.

Operation

The program is self-prompting and upon execution instructs the operator on what to do. Before running CLOAD 1, the TSC editor should be resident in RAM, and a disk file should be opened for later storage of the TRS-80 program. This is done under DOS 68.51 by typing, for example EDIT,, 80PROG.TXT. When the editor prompt appears, hit the computer reset switch and run CLOAD 1. The starting address is $B33B since the accompanying listing of CLOAD 1 is ORG'ed at $B000. If your system doesn't have memory here simply re-assemble it at the top of your memory map being careful to avoid conflicts with the editor or the DOS. The recorder is started as instructed by CLOAD 1, and the tape dump begins. CLOAD 1 waits for the sync pulse leader and uses 80 of the 128 Ø data bits to set up a software timing gate which will be used to search for later data bits. Use of such a software adaptive technique reduces loading problems due to recorder motor speed variations. Next, the program searches for the tape sync byte. If it is not properly decoded, a load error message is output to the terminal. The recorder volume control should be re-adjusted, and CLOAD 1 should be restarted. After locating the sync byte, the beginning and ending load address bytes are eaten, and program storage commences. The only wrinkle here is that level I line numbers are stored on tape in binary whereas the text editor will expect them to be in ASCII along with the program data. To allow room for later conversion; before being stored in RAM, the beginning of each BASIC line is padded with spaces which will eventually contain the ASCII equivalent of the line number. The program is stored in RAM starting at the address pointed to by memory location $0074 which is the TSC editor beginning-of-file buffer pointer. (FILBGN within CLOAD 1 must be changed to $0097 for SSB's SE-1 editor version and to $0098, I believe, for the FLEX 2.0 version.) When the carriage return of the last line is loaded, the running checksum is compared with the one on tape, and a load status message is issued from the terminal. The program automatically loads FILEND, the end-of-file buffer pointer with the correct ending

address of the program in RAM. (Again, CLOAD 1's FILEND must be changed to 80099 for the Smoke SE-1 version and to $009A for FLEX2.) The TRS-80 file must now be viewed or listed by performing a Jump to an address as instructed by the terminal. During this listing the binary line numbers are converted to their 5 digit ASCII equivalent. Any further listings, however, must be done through the editor which is next warm started by Jumping to location $0103. When this is done you will see that the TRS-80 program magically resides within the editor's text buffer for editing or SAVEing to disk in the BOPROG.TXT file opened earlier.

If you look closely at the listing of CLOAD 1 you will notice several seemingly useless NOP's and extraneous jumps. Although some loosness of code is certainly due to my novice programming ability, much of it is done in order to equalize the time variations through various program paths. The next logical step in this development is to modify the program to load and convert level II tapes. This will open the door to an even bigger store of software. The pulse timing relationships are different, but most of the software gates are adaptive and they will correct themselves. Furthermore, the hardware interface is designed to handle either level I or level II tapes. The only real complication is that level II program commands are stored in token form, and so conversion to editor compatible ASCII is necessary. If enough interest is shown, CLOAD2 will be the topic of a future article.

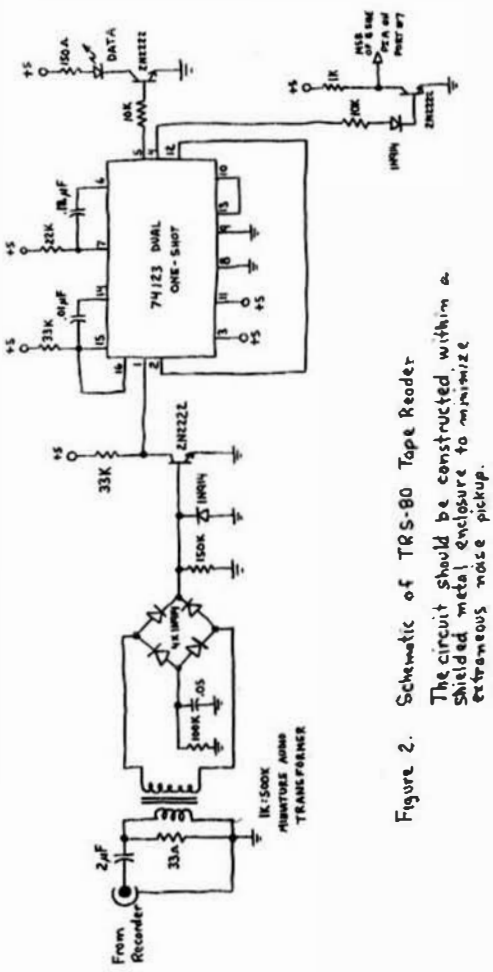


Figure 2. Schematic of TRS-80 Tape Reader
The circuit should be constructed within a shielded metal enclosure to minimize extraneous noise pickup.

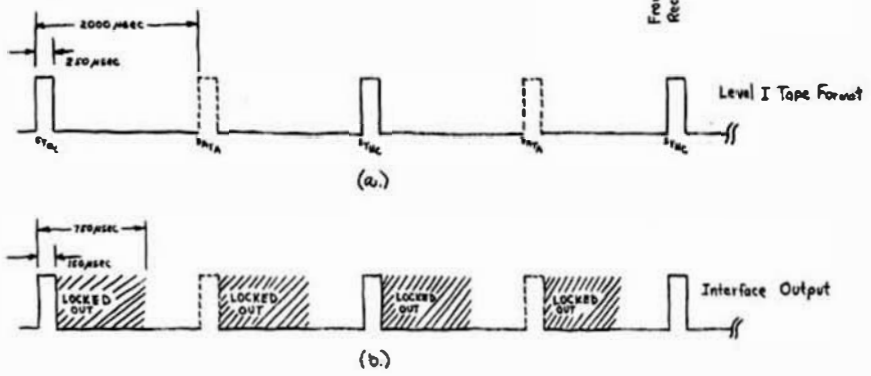


Figure 1 (a) TRS-80 LevelI BASIC tape format
(b) Response of tape reader interface to tape

```
CLOAD1                    6809 MNEMONIC ASSEMBLER  PAGE 1

                    1:        NAM   CLOAD1
                    2: *********************************************************
                    3: *                     CLOAD1                            *
                    4: *  A TRS-80 LEVEL I TAPE READER PROGRAM FOR THE 6800 MICRO  *
                    5: *                  TERRY RAYMOND                        *
                    6: *********************************************************

                   81 use  TIMING LOOPS SET UP FOR A 1.3 MHZ CPU CLOCK  ***

B000               10:        ORG   $B000
B000               11: SYNCNO RMB   1          NUMBER OF RECEIVED SYNC PULSES
B001               12: PERIOD RMB   1          AVG SYNC PULSE PERIOD
B002               13: SNCBYT RMB   1          TRS80 SYNC BYTE RECEIVED FLAG
B003               14: BYTREG RMB   1          REGISTER FOR SERIAL/PARALLEL CONV.
B004               15: WAIT   RMB   1          DELAY FROM SYNC END TO BIT PULSE
B005 1B            16: LOOK   FCB   $1B        TIME TO SPEND LOOKING FOR BIT PULSE
B006               17: BEGIN  RMB   1          IF $FF THEN 1ST '1' BIT NOT YET RCVD
B007               18: TEMPIX RMB   2          TEMPORARY
B009               19: SNCEND RMB   2          LOC OF LAST BYTE OF SOURCE CODE
0074               20: FILBGN EQU   $0074      TRSDOS DOS$0.5: EDITOR FILE BEGIN
0076               21: FILEND EQU   $0076      $0C/$0D DOS$0.5: EDITOR FILE END+1
B00B               22: POINTH RMB   1          MSB LINE POINTER
B00C               23: POINTL RMB   1          LSB LINE POINTER
B00D               24: TEMPA  RMB   1          TEMPORARY
B00E               25: TEMPB  RMB   1          TEMPORARY
B00F               26: ENDADR RMB   2          ADDRESS OF LAST BYTE LOADED
B011               27: TMPIXR RMB   2          TEMPORARY
B013               28: BUFFER RMB   5          BUILDS DEC. EQUIV OF EACH LINE#
B018 04            29: BUFEND FCB   $04
E0E3               30: CONTRL EQU   $E0E3
B019               31: TEMP1  RMB   2          TEMPORARY
B01B               32: DVDH   RMB   2          BINARY-> DECIMAL VARIABLE
B01D               33: DVSR   RMB   2          BINARY-DECIMAL VAR
B01F               34: TEMPC  RMB   2          TEMPORARY
B021               35: RMNDR  RMB   1          BINARY->DECIMAL VAR
B022               36: TEMPY  RMB   2          IX STORE DURING EOL CHECK
B024               37: TEMPZ  RMB   2          TEMPORARY
B026               38: STRSUM RMB   1          4 BYTE COUNTER FOR LEADING ADRS
B01E               39: PIA1BD EQU   $B01E      PORT B? IS THE INPUT PORT
B01F               40: PIA1BC EQU   PIA1BD+1
E07E               41: PDATA  EQU   $E07E
E1AC               42: INEEE  EQU   $E1AC
E1D1               43: OUTEEE EQU   $E1D1
E0CB               44: OUTAMS EQU   $E0CB
E047               45: BADDR  EQU   $E047
B027               46: STRADR RMB   2          START ADDRESS FOR TAPE DUMP
B029 00            47: CHKSUM FCB   $00        CHECKSUM ACCUMULATOR
B02A 0C            48: PULWID FCB   $0C        EFFECTIVE WIDTH OF BIT PULSE (150 us)
```

```
B02B 10              BANNER FCB  $10,$16,$0D,$  ,$0A,$00,$00
B032 43                     FCC  /(LOADB) ... a TRS-80 Level-I tape reader /
B03A 70                     FCC  /program for the 6800 microcomputer/
B07C 0A                     FCB  $0D,$0A,$0D,$00,$00
B08F 54                     FCC  /The TSC editor must be presently resident/
B0AA 20                     FCC  / along with an unused text file before/
B0D0 0D                     FCB  $0D,$0A
B0D2 44                     FCC  /dumping the tape. If this is true, then start /
B100 72                     FCC  /recorder and hit return./
B118 0D                     FCB  $0D,$0A,$07,$04

B11C 4C              OVRFL  FCC  /LOAD ERROR ... overflow in sync/
B13D 70                     FCC  /pulse period register/
B150 0D                     FCB  $0D,$0A,$07,$04
B154 4C              SUMERR FCC  /LOAD ERROR ... checksum did not verify/
B17A 0D                     FCB  $0D,$0A,$07,$04
B17E 4C              SNCERR FCC  /LOAD ERROR ... sync byte not located on tape/
B1AA 0D                     FCB  $0D,$0A,$07,$04
B1AE 0D              RETURN FCB  $0D,$0A,$0A,$04
B1B2 4C              SUMOK  FCC  /LOAD OK ... checksum did verify/
B1BL 0D                     FCB  $0D,$0A,$0A,$07,$04
B1D6 50              VIEWIN FCC  /Perform a Jump to location $/
B1F2 04                     FCB  $04
B1F3 70              RESTO  FCC  /to view or list the TRS-80 file./
B213 0D                     FCB  $0D,$0A,$00,$00
B217 44                     FCC  /During listing, this file will be automatically /

B247 63                     FCC  /converted to a TSC text editor /
B266 0D                     FCB  $0D,$0A,$00,$00
B26A 66                     FCC  /file which will reside/
B280 20                     FCC  / in the editor's text buffer and/
B2A0 20                     FCC  / be ready for editing./
B2BA 0D                     FCB  $0D,$0A,$00,$00
B2BA 0A                     FCB  $0A,$0D,$00,$00,$04
B2BF 0D              BOMFIN FCB  $0D,$0A,$0A,$00,$00
B2C4 54                     FCC  /This TSC compatible text file now resides from $/

B2F4 04                     FCB  $04
B2F5 74              BOMF10 FCC  /through $/
B2FE 04                     FCB  $04
B2FF 2E              LEAVE  FCB  $2E,$0A,$0D,$00,$00
B304 4A                     FCC  /Jump to location $0103 to restart the editor./
B331 0A                     FCB  $0A,$0D,$00,$00,$04
B336 0D              BT2    FCB  $0 ,$0A,$00,$00,$04

B33B 8E A042         START  LDS  #$A042     LOCATE STACK

     * CONFIGURE PIA FOR B-SIDE 7TH BIT AS BIT STREAM INPUT

B33E 7F 801E                CLR  PIA1BD
B341 7F 801F                CLR  PIA1BC
B344 86 7F                  LDA A #%01111111
B346 B7 801E                STA A PIA1BD
B349 86 04                  LDA A #%00000100
B34B B7 801F                STA A PIA1BC

B34E 86 32                  LDA A #50        NUMBER SYNC PULSES USED FOR TIMING
B350 B7 B000                STA A SYNCNO
B353 7F B001                CLR  PERIOD
B356 7F B002                CLR  SNCBYT
B359 7F B006                CLR  BEGIN
B35C 7F B003                CLR  BYTREG
B35F 7F B029                CLR  CHKSUM
B362 86 04                  LDA A #$04
B364 B7 B026                STA A STRSUM
B367 DE 74                  LDX  FILBGN
B369 FF B027                STX  STRADR      LOAD FILE INTO THE EDITOR

B36C CE B02D                LDX  #BANNER
B36F BD E07E                JSR  PDATA
B372 BD E1AC                JSR  INEEE
B375 CE B1AE                LDX  #RETURN

B378 BD E07E                JSR  PDATA

B37B B6 801E         LOOP1  LDA A PIA1BD
B37E 84 80                  AND A #%10000000
B380 27 F9                  BEQ  LOOP1       WAIT FOR FIRST SYNC PULSE
B382 B6 801E         STILUP LDA A PIA1BD
B385 84 80                  AND A #%10000000
B387 26 F9                  BNE  STILUP      FOUND IT, WAIT TILL OVER
B389 7D B000         MAXNUM TST  SYNCNO      ENOUGH SYNC PULSES YET?
B38C 27 30                  BEQ  READY       YES READY FOR DATA
B38E 5F                     CLR B
B38F 0C                     CLC
B390 5C              STILDN INC B            MEASURE NEW SYNC PERIOD
B391 8D B3AC                JSR  CHECK1
B394 B6 801E                LDA A PIA1BD
B397 84 80                  AND A #%10000000
B399 01                     NOP
B39A 27 F4                  BEQ  STILDN      WAIT FOR END OF THIS PERIOD
B39C 4F              UPDATE CLR A            NEW PERIOD IS OVER
B39D FB B001                ADD B PERIOD     UPDATE AVG SYNC PERIOD
B3A0 89 00                  ADC A #$00
B3A2 46                     ROR A
B3A3 56                     ROR B
B3A4 F7 B001                STA B PERIOD
B3A7 7A B000                DEC  SYNCNO
B3AA 20 D6                  BRA  STILUP
B3AC 25 01           CHECK1 BCS  OVRFLO
B3AE 39                     RTS
B3AF CE B11C         OVRFLO LDX  #OVRFL
B3B2 BD E07E                JSR  PDATA
B3B5 7E E0E3                JMP  CONTRL
B3B8 F6 B001         READY  LDA B PERIOD
B3BB 54                     LSR B
B3BC F0 B02A                SUB B PULWID
B3BF F7 B004                STA B WAIT
B3C2 73 B006                COM  BEGIN
B3C5 FE B027                LDX  STRADR
B3C8 08                     INX
B3C9 08                     INX
B3CA 86 01           NXTBYT LDA A #$01       SET END OF BYTE FLAG
B3CC B7 B003                STA A BYTREG
B3CF FF B022                STX  TEMPF       STORE INDEX REG
B3D2 CE 0400                LDX  #$0400      LOAD TIME-OUT TIMER
B3D5 09              LOOP2  DEX
B3D6 27 4D                  BEQ  ISLE1       MUST BE END OF LOAD
B3D8 B6 801E                LDA A PIA1BD     WAIT FOR SYNC PULSE
B3DB 84 80                  AND A #%10000000
B3DD 27 F6                  BEQ  LOOP2       NOT FOUND YET
B3DF FE B022                LDX  TEMPF       GOT IT,RESTORE INDEX REG
B3E2 B6 801E         STLUP  LDA A PIA1BD     FOUND ONE,WAIT TILL OVER
B3E5 84 80                  AND A #%10000000
B3E7 26 F9                  BNE  STLUP       STILL THERE
B3E9 01                     NOP
B3EA 01                     NOP
B3EB 01                     NOP
B3EC 20 01                  BRA  DELAY2
B3EE 39              DELAY1 RTS
B3EF B6 B004         DELAY2 LDA A WAIT       DELAY THE 'WAIT' TIME
B3F2 4A              LOOP3  DEC A
B3F3 8D F9                  BSR  DELAY1

B3F5 8D F7                  BSR  DELAY1
B3F7 26 F9                  BNE  LOOP3
B3F9 F6 B005                LDA B LOOK
B3FC 01                     NOP
B3FD 01                     NOP
B3FE 01                     NOP
B3FF 7E B402                JMP  TSTBIT      EQUALIZATION DELAY
B402 B6 801E         TSTBIT LDA A PIA1BD     WAIT IS OVER, LOOK NOW
B405 84 80                  AND A #%10000000
B407 26 0B                  BNE  FNDBIT      FOUND A DATA BIT
B409 01                     NOP
B40A 8D E2                  BSR  DELAY1
B40C 5A                     DEC B            LOOK FOR 'LOOK' TIME
B40D 26 F3                  BNE  TSTBIT
B40F 7D B006                TST  BEGIN       TEST FOR 1ST '1' BIT
B412 26 C1                  BNE  LOOP2       SYNC BYTE NOT IN YET
B414 7F B006         FNDBIT CLR  BEGIN       1ST '1' BIT FOUND
B417 48                     ASL A            MOVE BIT TO CARRY
B418 79 B003                ROL  BYTREG      ROTATE CARRY TO BYTREG LSB
B41B 25 0A                  BCS  CHECK       LAST BIT IS BYTE?
B41D 86 29                  LDA A #41        PULSE (250uS) DELAY
B41F 4A              LOOP4  DEC A
B420 26 FD                  BNE  LOOP4
B422 7E B3D5                JMP  LOOP2       GET NEXT BIT IN BYTE
B425 20 5E           ISLE1  BRA  SUMCHK
B427 7D B002         CHECK  TST  SNCBYT      RCVD SYNC BYTE YET?
B42A 26 12                  BNE  STORE
B42C F6 B003                LDA B BYTREG
B42F C1 A5                  CMP B #$A5       IS IT THE SYNC BYTE?
B431 26 49                  BNE  ERR2
B433 73 B002                COM  SNCBYT
B436 86 29                  LDA A #41        PULSE (250us) DELAY
B438 4A              LOOP5  DEC A
B439 26 FD                  BNE  LOOP5
B43B 7E B3CA                JMP  NXTBYT
B43E 7D B026         STORE  TST  STRSUM
B441 27 0A                  BEQ  STORE1
B443 7A B026                DEC  STRSUM
B446 B6 B003                LDA A BYTREG
B449 A7 00                  STA A 0,X
B44B 20 26                  BRA  PASSUP
B44D B6 B003         STORE1 LDA A BYTREG
B450 A7 00                  STA A 0,X
B452 BB B029                ADD A CHKSUM     UPDATE CHECKSUM
B455 B7 B029                STA A CHKSUM
B458 B6 B003                LDA A BYTREG
B45B 81 0D                  CMP A #$0D       WAS IT A CARRIAGE RTN?
B45D 26 14                  BNE  PASSUP      NO, JUST STORE THE BYTE
B45F 08                     INX
B460 86 00                  LDA A #$00
B462 A7 00                  STA A 0,X        PAD BEGINNING OF EACH NEW
B464 08                     INX              LINE WITH 4 MEMORY LOCATIONS
B465 A7 00                  STA A 0,X        FOR LATER NEEDED LINE #'S
B467 08                     INX
B468 A7 00                  STA A 0,X
B46A 08                     INX
B46B A7 00                  STA A 0,X
B46D 08                     INX
B46E A7 00                  STA A 0,X
B470 08                     INX
B471 A7 00                  STA A 0,X
B473 08              PASSUP INX
B474 86 1E                  LDA A #30
B476 4A              LOOP6  DEC A
B477 26 FD                  BNE  LOOP6
B479 7E B3CA                JMP  NXTBYT
B47C CE B17E         ERR2   LDX  #SNCERR
B47F BD E07E                JSR  PDATA
B482 7E E0E3                JMP  CONTRL
B485 FE B022         [illegible] LDX  TEMPF
B488 09                     DEX
B489 09                     DEX
B48A 09                     DEX
B48B 09                     DEX
B48C 09                     DEX
B48D 09                     DEX
B48E 09                     DEX
B48F 09                     DEX
B490 FF B009                STX  SRCEND
B493 08                     INX
B494 08                     INX
B495 08                     INX
B496 08                     INX
B497 08                     INX
B498 08                     INX
B499 08                     INX
B49A FF B00F                STX  ENDADR
B49D B6 B029                LDA A CHKSUM
B4A0 27 09                  BEQ  GOOD        BYTE SUM+2'S COMPL CHKSUM MUST = 0
B4A2 CE B154                LDX  #SUMERR
B4A5 BD E07E                JSR  PDATA
B4A8 7E E0E3                JMP  CONTRL
B4AB B6 B002         GOOD   LDA A SNCBYT
B4AE 27 CC                  BEQ  ERR2
B4B0 CE B1B2                LDX  #SUMOK
B4B3 BD E07E                JSR  PDATA
B4B6 CE B1D6                LDX  #VIEWIN
B4B9 BD E07E                JSR  PDATA
B4BC CE B4D1                LDX  #VIEWIT
B4BF FF B024                STX  TEMPZ
B4C2 CE B024                LDX  #TEMPZ
B4C5 BD E0C8                JSR  OUT4HS
B4C8 CE B1F3                LDX  #RESTO
B4CB BD E07E                JSR  PDATA
B4CE 7E E0E3                JMP  CONTRL

     ***** LIST AND CONVERSION ROUTINE *****

B4D1 CE B1AE         VIEWIT LDX  #RETURN
B4D4 BD E07E                JSR  PDATA
B4D7 FE B027                LDX  STRADR      LOCATION OF 1st BYTE OF LOAD
B4DA 08                     INX
B4DB 08                     INX
B4DC 08                     INX
B4DD 08                     INX
B4DE 08                     INX
B4DF 08                     INX
B4E0 E6 00           LINENO LDA B 0,X        GET LSB OF BINARY LINE #
B4E2 08                     INX
B4E3 A6 00                  LDA A 0,X        GET MSB OF BINARY LINE #
B4E5 FF B011                STX  TMPINX
B4E8 CE B013                LDX  #BUFFER
B4EB BD B5A9                JSR  BNTODC      GET DECIMAL EQUIV
B4EE FE B011                LDX  TMPINX      NOW GO BACK AND CONVERT
B4F1 B6 B017                LDA A BUFFER+4   THE LINE #'S TO 5 DIGIT
B4F4 A7 00                  STA A 0,X        BCD ASCII EQUIV
B4F6 09                     DEX
B4F7 B6 B016                LDA A BUFFER+3
B4FA A7 00                  STA A 0,X
B4FC 09                     DEX
B4FD B6 B015                LDA A BUFFER+2
B500 A7 00                  STA A 0,X
B502 09                     DEX
B503 B6 B014                LDA A BUFFER+1
B506 A7 00                  STA A 0,X
B508 09                     DEX
B509 B6 B013                LDA A BUFFER
B50C A7 00                  STA A 0,X
B50E CE B013         BACK   LDX  #BUFFER
B511 BD E07E                JSR  PDATA       OUTPUT THE DEC. STRING
B514 FE B011                LDX  TMPINX      RETRIEVE INDEX REG
B517 08              GETCHR INX
B518 BC B009                CPX  SRCEND      LAST SOURCE BYTE?
B51B 27 20                  BEQ  FINISH
B51D A6 00                  LDA A 0,X
B51F 81 0D                  CMP A #$0D
B521 26 15                  BNE  OUTCHR
B523 BD E1D1                JSR  OUTEEE
B526 86 0A                  LDA A #$0A
B528 BD E1D1                JSR  OUTEEE
B52B 86 00                  LDA A #$00
B52D BD E1D1                JSR  OUTEEE
B530 86 00                  LDA A #$00
B532 BD E1D1                JSR  OUTEEE
B535 08                     INX
B536 20 A2                  BRA  LINENO-6
B538 BD E1D1         OUTCHR JSR  OUTEEE
B53B 20 DA                  BRA  GETCHR
B53D 7F B00B         FINISH CLR  POINTH
B540 7F B00C                CLR  POINTL
B543 FE B027                LDX  STRADR
B546 09                     DEX
B547 08              GOIT   INX
B548 86 01                  LDA A #$01
B54A 5F                     CLR B
```

```
B54B BB B00C  345        ADD A POINTL
B54E 19       346        DAA
B54F B7 B00C  347        STA A POINTL
B552 A7 01    348        STA A 1,X
B554 17       349        TBA
B555 B9 B00B  350        ADC A POINTH
B558 19       351        DAA
B559 B7 B00B  352        STA A POINTH
B55C A7 00    353        STA A 0,X
B55E 08       354        INX
B55F 08       355 SEARCH INX
B560 BC B009  356        CPX   SRCEND
B563 27 08    357        BEQ   COMPLT
B565 A6 00    358        LDA A 0,X
B567 81 00    359        CMP A #$00
B569 26 F4    360        BNE   SE
B56B 20 DA    361        BRA   GOT1
B56D FE B027  362 COMPLT LDX   STRADR
B570 08       363        INX
B571 08       364        INX
B572 86 00    365        LDA A #$00
B574 A7 00    366        STA A 0,X
B576 CE B2BF  367        LDX   #DONEIN
B579 BD E07E  368        JSR   PDATA
B57C FE B027  369        LDX   STRADR
B57F FF B024  370        STX   TEMP2
B582 CE B024  371        LDX   #TEMP2
B585 BD E0C8  372        JSR   OUT4HS
B588 FE B009  373        LDX   SRCEND
B58B 08       374        INX
B58C DF 76    375        STX   FILEND
B58E CE B2F5  376        LDX   #DONEIO
B591 BD E07E  377        JSR   PDATA
B594 FE B009  378        LD    SRCEND
B597 FF B024  379        STX   TEMP2
B59A CE B024  380        LDX   #TEMP2
B59D BD E0C8  381        JSR   OUT4HS
B5A0 CE B2FF  382        LDX   #LEAVE
B5A3 BD E07E  383        JSR   PDATA
B5A6 7E E0E3  384        JMP   CONTRL

B5A9 CE B012  386 BNTODC LDX   #BUFFER
B5AC FF B019  387        STX   TEMP1
B5AF 08       388        INX
B5B0 08       389        INX
B5B1 08       390        INX
B5B2 08       391        INX
B5B3 B7 B01B  392        STA A DVDN
B5B6 F7 B01C  393        STA B DVDN+1
B5B9 86 0A    394        LDA A #10
B5BB B7 B01D  395        STA A DVSR
B5BE FF B01F  396 LOOP9  STX   TEMPC
B5C1 BD B5DD  397        JSR   UNSPD             DIVISION ROUTINE
B5C4 FE B01F  398        LDX   TEMPC
B5C7 B6 B021  399        LDA A RMNDR
B5CA 8B 30    400        ADD A #$30
B5CC A7 00    401        STA A 0,X
B5CE 09       402        DEX
B5CF 8C B012  403        CPX   #BUFFER-1
B5D2 26 EA    404        BNE   LOOP9
B5D4 FE B019  405        LDX   TEMP1
B5D7 08       406        INX
B5D8 08       407        INX
B5D9 08       408        INX
B5DA 08       409        INX
B5DB 08       410        INX
B5DC 39       411        RTS
B5DD 7F B021  412 UNSPD  CLR   RMNDR
B5E0 C6 11    413        LDB   #17
B5E3 20 08    414        BRA   START1
B5E5 B6 B021  415 LOOP10 LDA A RMNDR
B5E8 B0 B01D  416        SUB A DVSR
B5EB 2A 03    417        BPL   NREST
B5ED 0C       418 START1 CLC
B5EE 20 04    419        BRA   NERDO
B5F0 B7 B021  420 NREST  STA A RMNDR
B5F3 0D       421        SEC
B5F4 79 B01C  422 NERDO  ROL   DVDN+1
B5F7 79 B01B  423        ROL   DVDN
B5FA 09       424        DEX
B5FB 27 05    425        BEQ   RTN
B5FD 79 B021  426        ROL   RMNDR
B600 20 E3    427        BRA   LOOP10
B602 39       428 RTN    RTS
              429        END
    NO ERROR(S) DETECTED
```

SYMBOL TABLE:

BACK	B54E	[illegible]	E047	BAADWN	B389	[illegible]	B02D
BEGIN	B000	BNTODC	B5A9	BUFEND	B01B	BUFFER	B012
BYTREG	B003	CHECK	B437	CHECK1	B3AC	CHKSUM	B029
COMPLT	B56D	CONTRL	E0E3	DELAY1	B3EE	DELAY2	B3E7
DONEIN	B2BF	DONEIO	B2F5	DVDN	B01B	DVSR	B01D
ENDADR	B00F	ERR2	B47C	FILBGN	0074	FILEND	0076
FINISH	B53D	FNDBIT	B414	GETCHR	B517	GOOD	B4AB
GOT1	B547	INEEE	E1AC	ISLE1	B420	LEAVE	B2FF
LINEND	B4E0	LOOK	B00	LOOP1	B379	LOOP10	B5E5
LOOP	B385	LOOP3	B3F7	LOOP4	B41F	LOOP8	B438
LOOP6	B474	LOOP9	B5BE	NERDO	B5F4	NREST	B5F0
NXTBYT	B3CA	OUT4HS	E0C8	OUTCHR	B538	OUTEEE	E1D1
OVRFL	B11C	OVRFLO	B3AF	PASSUP	B473	PDATA	E07E
PERIOD	B001	PIA1BC	B01F	PIA1BD	B01E	POINTH	B00B
POINTL	B00C	PULWID	B02A	READY	B388	RESTO	B1F3
RETURN	B14E	RMNDR	B021	RT2	B33A	RTN	B602
SEARCH	B55F	SNCBYT	B002	[illegible]	B17E	SRCEND	B009
START	B33B	START1	B5ED	STILUP	B390	STLUP	[illegible]
STLUP	B2E2	STORE	B43E	STORE1	B44D	STRADR	B027
STRSUM	B02	[illegible]	B485	SUMEND	B154	SUMCK	B1B2
SYNCNO	B000	TEMP1	B019	TEMPA	B00D	TEMPB	B00E
TEMPC	B01F	TEMPF	B022	TEMPIX	B007	TEMP2	B024
TMPIXR	B011	TSTBIT	B402	UNSPD	B5DD	UPDATE	B39C
VIEWIN	B1D6	VIEWIT	B4D3	WAIT	B004		

TRS-80 is a Trademark of the Tandy Corporation.

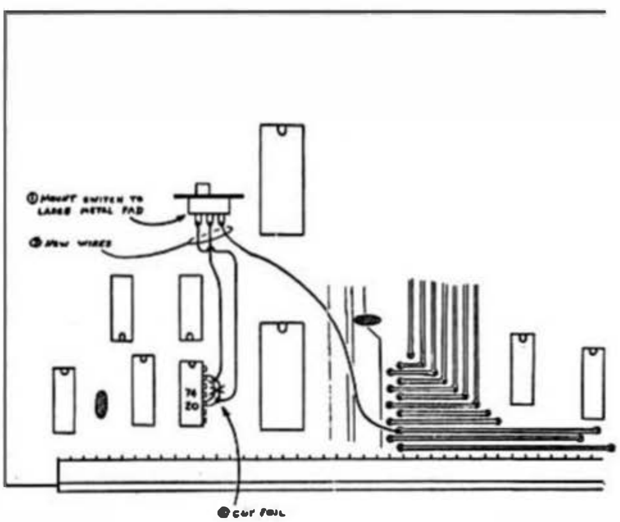


Figure 1: Modification for MP-A board.

Diagrams Continued from Page 17.

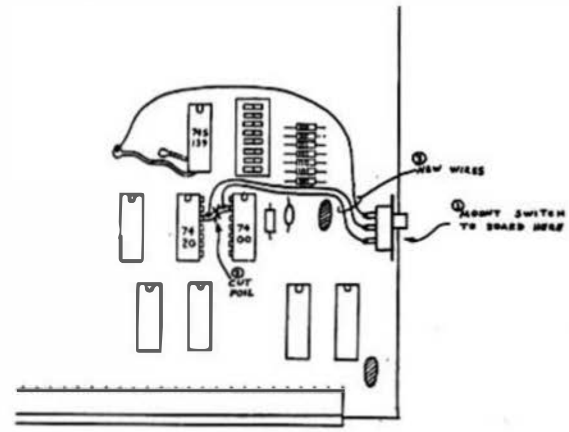
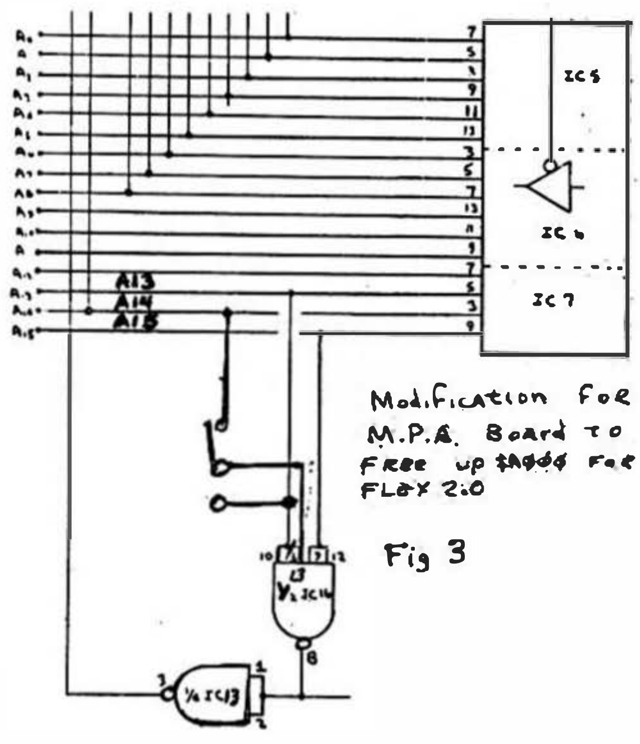
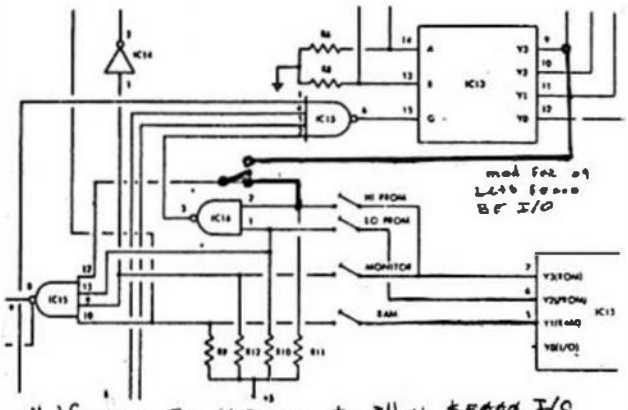


Figure 2: Modification for MP-A2 board





Modification For M.P.A-2 to allow $E800 I/O

Fig 4

BIT Bucket

Rayward Sneed
1665 14th St. 8203
Boulder, Colorado 80302

October 20, 1980

Mr. Don Williams
'68' Micro Journal
3018 Hamill Rd.
Hixon, TN 37343

Dear Mr. Williams!

I would like to relate an excellent experiences with Lucidata and their fine Pascal system for the M6800/09. I was not enthusiastic about purchasing such a complex software system from distant Holland, but my desire to have a Pascal which would run under an existing general-purpose operating system (FLEX/9) outweighed my initial fears.

I recieved the package promptly, consisting of a well-written manual and a distribution disk containing the system and a series of excellent demonstration programs. After reading the manual, I proceeded to compile and execute some of the more interesting demos. The system performed as advertised, until I attempted to compile a program which included a function declaration. Further experimentation revealed that the system simply could not compile any program with a function in it. I decided that I must have a flawed copy of the distribution disk, and packed it off to Holland with a careful description of my problem.

After 2 or 3 weeks, I received a reply from the Nisel Bennee, the author of the Pascal run-time system. Nisel was unable to reproduce my problem, and suggested I check carefully for memory problems. My two diagnostic routines showed no problems, and I wrote an unpleasant letter to Lucidata demanding a solution.

To my utter amazement, after less than a week Nisel telephoned me from Holland. We discussed the problem, and he suggested that I make several tests and send him the results. This I did, and a few days later a card arrived informing me of the existence of another 6809 Pascal user in nearby Loveland, Colorado (our letters had crossing in the mail). The other system DID compile functions correctly, so I arranged to bring my machine to Loveland for some tests.

To make a long story short, Nisel (in Holland) and I (in Loveland) independantly discovered that the copy I was working from had ONE bit in error. I later found that the error had occured when I made a working copy from the distribution disk, and that it was caused by a bad memory location. I replaced the defective chip, and have been running smoothly ever since.

So Nisel was right all along. At any rate, I have never received such excellent support from any company, regardless of their location. Those of you hesitating about purchasing either version of Lucidata's Pascal system: hesitate no more. It is an excellent system, and they are serious about supporting it.

Sincerely,

Rayward Sneed

October 8,1980

'68' Micro Journal,
3018 Hamill Road.,
P.O.Box 849,
Mixson, Tennessee,
37343, U.S.A.

Dear Editor:-

EXTRA 4K MEMORY SPACE ON SWTPC MP-09 CPU CARD

There are three extra sockets provided on SWTPC MP-09 CPU card. They are intended for additional 2716 pin compatible memories, but as mentioned in the instruction manual, this address space conflicts with DMA controller of 8" disk driver or I/O port addresses. Chips cannot be pluged in simply. Some modifications on the mother board and CPU card are required to use the space.

Despite the fact that these modifications are somehow dirty and chips are still expensive, it is worth while doing. This additional space is quite useful for custom applications because they are isolated from the main memory back and never influenced by the main program such as basic. I use them for driver routines of Hazelwood's 250 x 250 videoboard, timer driver, etc.

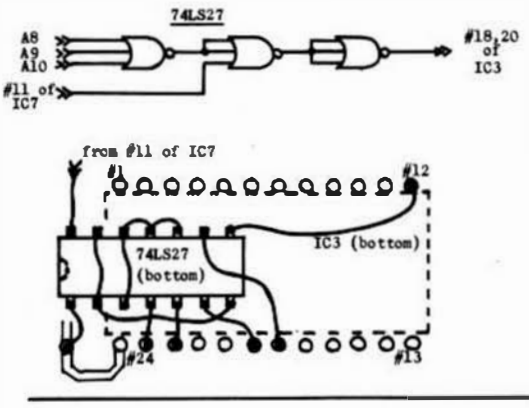
Yours truly,

Kunihiro Kitadera,
126 Sedgefield,
Pointe Claire,Quebec.
H9R 1N5, CANADA
514-694-1643

MODIFICATIONS

Modification on the mother board is well documented in Keith Alexander's article which appeared in the April '80 issue of Micro Journal. By this modification 2 K bites of memory $E800-$EFFF becomes available. Plus a 2716 or compatible RAM (SWTPC sells it) on second to the rightmost socket, turn dipswitch-3 on, and insert jumper plug to RAM or ROM depend on the chip selection.

Modification on CPU card is shown in the picture below. The idea of this modification is to disable memory addressing to $E800-$E8FF in order to avoid conflict with I/O address, and enable to use $E100-$E7FF. I glued a 74LS27 chip behind the rightmost socket and soldered directly to its pins. But anybody who does not want to touch CPU card could make small PROM carrier PC card and put it onto the socket. To activate this memory, select jumper plug position to RAM or ROM. Leave dipswitch-2 "off" this time. As mentioned above, address $E800-$E8FF of the memory is disabled and only $E100-$E7FF is usable.



CPU CARD MODIFICATION

JOHN TUCKER
POST OFFICE BOX 2898
LAREDO, TEXAS
78041
OCTOBER 12, 1980

DEAR DON,

SORRY, BUT THE IBM IS TIED UP RIGHT NOW. THIS IS THE ONLY PRINTER AVAILABLE.

THERE HAVE BEEN SEVERAL ARTICLES RECENTLY IN THE JOURNAL ABOUT TROUBLES WITH DISK SYSTEMS ERRORING OUT ON WRITES, ESPECIALLY FROM TEXT FILES. I OFFER TWO THINGS THAT HAVE TOTALLY CURED (AND I MEAN TOTALLY) MY DISK WRITE PROBLEMS. I HAVE TWO SHUGART 5 INCH DRIVES RUNNING UNDER MINIFLEX. BECAUSE OF AN ODDITY IN THE TEXT PROCESSOR THAT I USE, THE DRIVES MUST RUN CONTINUOUSLY WHILE TEXT PROCESSING IS BEING DONE. THESE DRIVES OFTEN, VERY OFTEN, RUN AS LONG AS FIVE HOURS WITHOUT THE MOTORS STOPPING AND WITH NEVER A READ ERROR.

OBVIOUSLY THESE FILES ARE CREATED WITHOUT A WRITE ERROR, ELSE THEY WOULD NEVER PRINT OUT CORRECTLY. THUS, I KNOW THAT MY DISK DRIVES ARE BOTH WRITING AND READING AS THEY SHOULD. IT IS NOT UNCOMMON TO SAVE A FILE 100 TO 120 SECTORS LONG, ALWAYS WITHOUT ERROR.

MY SYSTEM HAS BEEN AROUND QUITE A WHILE. THE DISK CONTROLLER CARD WAS ORIGINALLY A DC-1 CARD. IT WAS NECESSARY TO REPLACE THAT WITH A DC-2 CARD TO OBTAIN RELIABLE OPERATION. ALSO, THE MP-A CARD WAS REPLACED WITH AN MP-A2 CARD.

THAT IS WHEN THE FUN BEGAN. EVERYTHING FAILED THAT WAS TIME-RELATED. THE DISKS WROTE WHERE THEY WANTED TO, READ WHAT THEY WANTED TO, AND IN GENERAL THEY WENT NUTS. I WAS EVEN DOWN TO USING THE AC-30 AND THE CASSETTES FOR A WHILE, EVEN THOUGH THE PROBLEM WAS OBVIOUS. I HAD TIMING PROBLEMS RUNNING OUT OF MY EARS -- AND OTHER OPENINGS.

WHEN IN DOUBT, READ THE INSTRUCTIONS! I ASSEMBLED THE LITTLE TIMING PROGRAM THAT CAME WITH THE MP-A2 CARD, RAN IT, AND THERE WAS MY PROBLEM. THE PROCESSOR WAS LOAFING ALONG SOMEWHERE DOWN AROUND 730 KHZ. A LITTLE TRIMMING OF THE APPROPRIATE RESISTOR (NOTED IN THE INSTRUCTIONS) AND THE FREQUENCY ROSE TO 900 KHZ. DISK ERRORS INSTANTLY DROPPED TO ZERO. A 4 MHZ CRYSTAL WAS OBTAINED AND SUBSTITUTED FOR THE CAPACITOR IN THE TIMING CIRCUIT BRINGING THE PROCESSOR TIMING TO EXACTLY 1000 KHZ (1 MHZ).

SINCE THIS CORRECTION WAS MADE, I HAVE NOT HAD A SINGLE DISK READ OR WRITE FAILURE. I HAVE HAD ONE DISK FAIL MECHANICALLY (IT JAMMED IN ITS HOLDER AND CANNOT BE TURNED, EVEN BY HAND). LUCKILY, A BACKUP OF IT EXISTED AND THE NEW FILE STILL REMAINED IN THE TEXT EDITOR SO ALL OF THE DATA WAS SAFE.

RATHER THAN DASH OFF IN SEVERAL DIRECTIONS AT ONCE, GRABBING TEST GEAR AS YOU GO, USE YOUR SOFTWARE TO TEST YOUR SYSTEM BEFORE YOU START BLAMING THE HARDWARE. MY PROBLEMS WERE ALL DUE TO A RESISTOR THAT HAD THE WRONG COLOR BANDS PAINTED ON IT -- IT SAID IT WAS 470 OHMS BUT IT MEASURED WAY UP AROUND 620. A SIMPLE 1000 OHM RESISTOR SHUNTED ACROSS IT SOLVED MY PROBLEMS. MAYBE YOUR PROBLEMS ARE AS SIMPLE.

SECONDLY, AND DON'T ASK ME WHY, IN SAVING A FILE TO DISK FROM THE TSC TEXT EDITOR, TYPE A 'T' (GO TO TOP OF FILE) BEFORE TYPING THE 'S' TO SAVE THE FILE TO DISK. I HAVE NO IDEA ON EARTH WHY THIS HELPS, BUT IT DOES.

BUT ALWAYS, REMEMBER TUCKER'S COROLLARY OF MURPHY'S LAW: EVERY SILVER LINING HAS A DARK CLOUD.

BEST WISHES,

John

JOHN TUCKER

GIMIX, Inc.
1337 West 37th. Place
Chicago, Illinois 60609

Don Williams
68 Micro Journal
3018 Hamill Road
Hixson, Tennessee 37343

Dear Don,

This is in answer to your questions on the unique features of our GIMIX versions of FLEX for the 6809. This enhanced version of TSC's FLEX is for the 5/8 controller card, and we will have other versions available for our forthcoming double density and DMA controller cards. All versions are compatible with programs written for use with FLEX09. Some of its features are:

- Runs any combination of up to four 5 and 8 inch disk drives.
- Runs single or double headed drives
- Auto size selection for systems with both 5 and 8 inch drives.
- Has software selection of stepping speed by drive.
- Print spooling using the 6840 on the GIMIX 6809 PLUS CPU Card.
- Format program can format single and double tracking 5 and 8 inch drives:
  - For 5 inch drives:
    - 35 and 40 tracks at 48 Tracks per inch
    - 77 tracks at 100 Tracks per inch
    - 80 tracks at 96 Tracks per inch
  - For 8 inch drives:
    - 77 tracks at 48 Tracks per inch
    - 154 tracks at 96 Tracks per inch
- Format prompts for the number of tracks to format (1 - 80 tracks for 5" and 1 - 154 tracks for 8").
- Format program has selectable interleave for 1 or 2MHz operation for optimized disk read/write speed.
- Format program can format disks in double tracking drives for use in single tracking drives.
- Enhanced 'Super Boot' has full error trapping including 'Disk Not Linked'.
- Utilities for Reading and Setting the Time of Day Clock option on the GIMIX 6809 CPU Card.
- Source code for all hardware support programs included on disk.
- No patches necessary to run on terminal based systems or systems with the GIMIX 80 X 24 video board and GMXBUG-09 with the GMXBUG-09 Video Drivers.
- Includes source and binary for the control/C patch for both of TSC's BASICS for use with the Video based version of GMXBUG-09.
- Boots up and runs under GMXBUG-09, SONEXBUG and SBUG-E.
- Runs at 2MHz with mother boards that have slow I/O.

Also available from GIMIX is a position independent Disk Boot PROM with full error trapping.

Sincerely,

Michael H. Katz

Note: FLEX (TM) is a trademark of Technical Systems Consultants, West Lafayette, Indiana. GIMIX (R) is a registered trademark of GIMIX, Inc., Chicago, Illinois.

UNIVERSAL DATA RESEARCH INC.
2361 WEHRLE DRIVE • BUFFALO, NEW YORK 14221 • 716-631-3011

October 17, 1980

UPDATE NOTICE

Contact: Joel W. Hackman
UNIVERSAL DATA RESEARCH
2457 Wehrle Drive
Buffalo, New York 14221
716-631-3011

Mr. Don Williams
68 MICRO JOURNAL
3018 Hamill Road
P.O. Box 849
Hixson, Tenn. 37343

Dear Don:

We would like to thank you and your magazine for the excellent reviews on our DATA BASE MANAGEMENT SYSTEM.

We would like to announce the release of Version 4.1 with the following enhancements:

TERMINAL INDEPENDENCE

SCREEN EDIT ROUTINES

MULTI FORMAT REPORT WRITER AND LABEL PRINTER with PARAMETER STORAGE

This update is on us! Any customer wishing to receive Version 4.1 should send back their original disk for an update at no charge.

Thanks Don, keep up the good work!

Sincerely,

UNIVERSAL DATA RESEARCH, INC.

Joel W. Hackman
President

JWH/jm

P.S. UNIFLEX Version will be finished shortly!

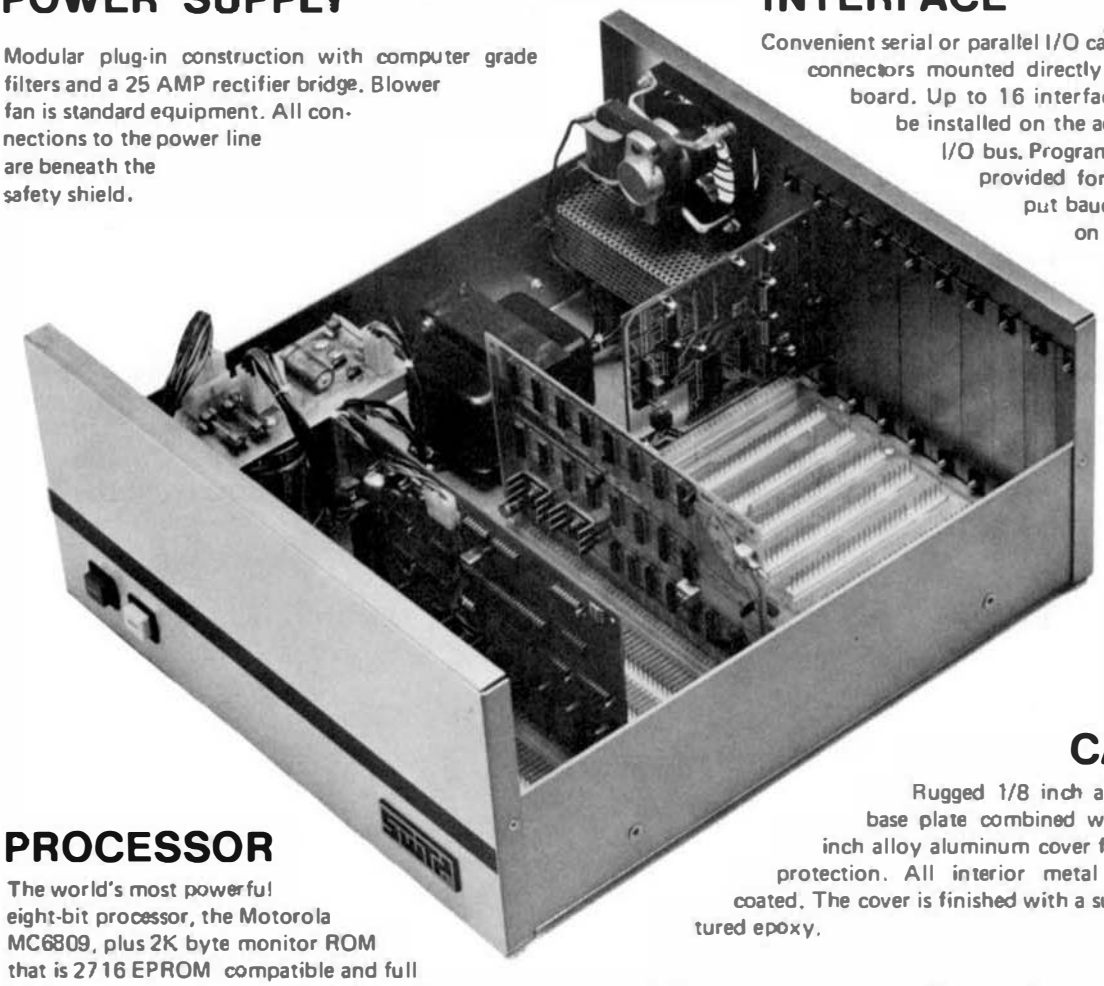
WE HAVE A 6809 FOR YOU

POWER SUPPLY

Modular plug-in construction with computer grade filters and a 25 AMP rectifier bridge. Blower fan is standard equipment. All connections to the power line are beneath the safety shield.

INTERFACE

Convenient serial or parallel I/O cards have DB-25 connectors mounted directly on the circuit board. Up to 16 interface devices may be installed on the address decoded I/O bus. Programming strips are provided for input and output baud rate selection on each port. All outputs are fully buffered.



CABINET

Rugged 1/8 inch alloy aluminum base plate combined with a solid 1/8 inch alloy aluminum cover for unsurpassed protection. All interior metal is conversion coated. The cover is finished with a super tough textured epoxy.

PROCESSOR

The world's most powerful eight-bit processor, the Motorola MC6809, plus 2K byte monitor ROM that is 2716 EPROM compatible and full buffering on all output lines. Built-in multiuser capability, just add I/O cards to operate a multi-terminal system.

MEMORY— You can purchase the computer with either 8K bytes of RAM memory (expandable to 56K), or with the full 56K. The efficient, cool running dynamic memory used in this system is designed and manufactured for us by "Motorola Memory Systems Inc."

PERIPHERALS—The wide range of peripheral hardware that is supported by the 6809 includes: dot matrix printers (both 80 and 132 column), IBM Electronic 50 typewriter, daisy wheel printers, 5-inch floppy disk system, 8-inch floppy disk systems and a 16 megabyte hard disk.

SOFTWARE— The amount of software support available for the 6809 is incredible when you consider that it was first introduced in June, 1979. In addition to the FLEX9 operating system, we have a Text Editor, Mnemonic Assembler, Debug, Sort-Merge, BASIC, Extended BASIC, MultiUser BASIC, FORTRAN, PASCAL and PILOT.

69/K Computer Kit with 8K bytes of memory ..... $ 495.00
69/A Assembled Computer with 8K bytes of memory ..... $ 595.00
69/56 Assembled Computer with 56K bytes of memory $1,595.00

SWTPC

SOUTHWEST TECHNICAL PRODUCTS CORPORATION
219 W. RHAPSODY
SAN ANTONIO, TEXAS 78216 (512) 344-0241

6809 DISK SYSTEMS

All disk systems are supplied with our version of FLEX 9, the world standard disk operating system for the 6809. Our systems normally operate in double density format, but they are compatible with single density, or single sided recording formats. FLEX is supplied with over forty utilities, many of which are only available with our systems.

Our disk systems offer you mass storage at low cost. The cost per thousand bytes of storage for our various systems is shown in the chart. Other 6809 disk systems have costs up to three times greater for the same general type drive.

TYPE	CAPACITY	COST
D-5	720,000 bytes	$1.80 per/K
DT-5	1,400,000 bytes	$1.16per/K
DMF-2	2,400,000 bytes	$1.04 per/K
CDS-1	16,000,000 bytes	$ .27 per/K

D-5 Two double sided, double density, 5" disk drives with a total on line capacity of 720,000 bytes of data. Includes cabinet, power supply, connecting cable and controller. Controller will operate up to four drives. This is an ideal disk system for small stand alone word processing systems, or for businesses that do not work with large inventories.

14 x 6 x 10 – 20 lbs $1,295.00

D-5 or DT-5

DT-5 Double track density version of the D-5. The DT-5 uses two 96 track per inch drives to provide an on line capacity of 1,400,000 bytes. Includes cabinet, power supply, connecting cable and controller. Controller will operate up to four drives. This is a disk system with enough capacity to include small inventories of up to 1,000 items, plus the usual business package of general ledger payroll, etc.

14 x 6 x 10 – 20 lbs $1,695.00

DMF2

DMF-2 Double sided, double density, dual eight-inch disk system with an on line capacity of 2,400,000 bytes. Our "top of the line" disk system features a DMA type controller for fastest possible data transfers. This drive was designed for larger businesses and multi user installations. The DMF-2 will provide the fast operation necessary for systems running multiterminals under the UniFLEX operating system. Complete with a heavy duty 1/8-inch metal cabinet, power supply, connecting cable and controller. The controller will operate up to four drives.

17½ x 5 x 21½ – 53 lbs $2,495.00

CDS-1 This "Winchester" type hard disk provides both large storage capacity and high speed operation. The CDS-1 is the answer for systems that must handle large inventories or systems with more than four terminals. The controller has its own processor and uses DMA data transfer.

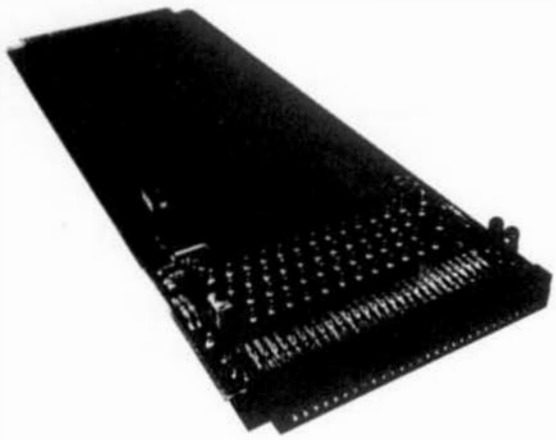
CDS-1 – 116 lbs $4,395.00

CDS-1

SOUTHWEST TECHNICAL PRODUCTS CORPORATION
219 W. RHAPSODY
SAN ANTONIO, TEXAS 78216 (512) 344-0241

Datricon STD BUS EXTENDER BD.

7911 N. E. 33rd Drive
Suite 200
Portland, Oregon 97211
503-284-6277



DATRICON CORPORATION NOW PROVIDES AN ALL NEW STD BUS EXTENDER CARD THAT PROVIDES THE USER WITH BASIC ACCESS TO SIGNALS. TROUBLESHOOTING OF STD BUS SYSTEMS IS GREATLY AIDED USING THIS EXTENDER WITH LOGIC ANALYZERS, OSCILLOSCOPES OR OTHER DEVICES THAT REQUIRE DIRECT BUS INTERFACES. A MUST FOR MAINTAINING ANY STD BUS SYSTEM.

POWER SUPPLY STATUS LAMPS:	Indicate all four supply voltages exceed their minimum requirements.
PUSHBUTTON RESET:	Pushbutton Reset Switch provides readily available access to STD BUS PBRESET* signal
PRIORITY CHAIN (IB-OUT):	Priority Chain Switch permits chaining signal to bypass the extender for times when board is installed only for purposes of monitoring STD BUS signals, without a card inserted in extender receptacle connector.
STANDARD BUS SIGNAL TERMINALS:	All STD BUS signals are identified by the appropriate mnemonic and a terminal for attaching logic analyzers, signal analyzers, oscilloscopes, etc.

FOR IMMEDIATE RELEASE

STD BUS Extender Board With Terminal Posts Simplifies System Tests

Datricon's STD BUS Extender Board provides support to in-service systems. All STD BUS signals are identified by the appropriate mnemonic and have a terminal post for attaching logic analyzers and oscilloscopes.

Extender Board meets dimensional requirements for STD BUS.

Power Supply status lamps indicate when each supply exceeds the minimum.

Push Button Reset Switch permits ready access to STD BUS PBRESET* signal.

Priority Chain Switch permits chaining signal to bypass Extender Board recepticle.

This STD BUS Extender Board is available from stock for $85 each.

SANSASKA SYSTEMS
3311 Concord Blvd
Concord, CA 94519

June, 1980 SAMBUG09

SANSASKA SYSTEMS announces SAMBUG09 - an improved 6809 Monitor program. It features: memory examination in 3 formats, vectoring of all interrupts, search for other modules, tape punch (S1 format), tape load with and without offset, device control blocks for I/O, memory search for masked match, 8 transparent breakpoints, register display and change, jump (JSR) and GO (RTI) to program, 16 bit add and subtract, abort, random number generator, memory checksum, memory copy, memory fill, RAM test, disk boot, and expandability to mnemonic memory display and change.

The program is compatable with FLEX (FLEX is copyright by Technical Systems Consultants) and is available in EPROM. SAMBUG09 is available in several configurations and on several types of EPROM to fit the multiplicity of 6809 systems currently appearing on the scene. Current versions will run on a modified SWTPC (described in the May 1980 issue of 68 MICRO), a Southwest Technical Systems 6809 CPU board, the Percom adapter board for the 6800 CPU board, and the Percom 6809 CPU board. Further customization is available.

From initialization SAMBUG09 will find your expansion module and initialize it, allowing you to add or change functions and commands. SAMBUG09 provides many EPROM resident routines that may be called to provide various types of I/O service and other useful functions.

SAMBUG09 is available from: SANSASKA SYSTEMS, 3311 Concord Blvd., Concord CA, 94519. (415) 676-2268.

SANSASKA SYSTEMS
3311 Concord Blvd
Concord, CA 94519

June, 1980 SAMBUG09

FEATURE	SAMBUG09	PSYMON	SBUG-E
MEMORY EXAMINE (NUMBER OF FORMATS)	3/4	1	1
INTERRUPT VECTORING	ALL	ALL	ALL BUT NMI
OTHER MODULE SEARCH	YES (3)	YES (1)	NO
TAPE PUNCH	YES	YES	YES
TAPE READ WITH OUT OFFSET	YES	YES	YES
TAPE READ WITH OFFSET	YES*	NO	NO
DCB'S FOR I/O	YES	YES	NO
MEMORY SEARCH FOR MATCH	YES	NO	NO
MEMORY SEARCH FOR MASKED MATCH	YES*	NO	NO
NUMBER OF BREAKPOINTS	8	10	5
TRANSPARENT	YES	NO	NO
REGISTER DISPLAY/CHANGE	YES	YES	YES
JUMP (JSR) TO PROGRAM	YES	YES	NO
GO (RTI) TO PROGRAM	YES	YES	YES
16 BIT ADD, SUBTRACT	YES	NO	NO
ABORT	YES	NO	NO
RANDOM NUMBER GENERATOR	YES	NO	NO
MEMORY CHECKSUM	YES*	NO	NO
COPY MEMORY	YES*	NO	NO
FILL MEMORY WITH ANY VALUE	YES*	NO	NO
TEST RAM	YES*	NO	YES
BOOT DISK	YES*	NO	YES
MNEMONIC EXPANSION CAPABILITY	YES	NO	NO
DISPLAY STACK	NO	NO	YES
CALL PROM ROUTINE AT C000	NO	YES	NO
SIZE	2/3K	1K	2K

*3K VERSION ONLY

FIGURE 1. Comparison of the features of 3 6809 monitors.

NEW PRODUCTS RELEASE

Cincitex Software is pleased to announce the availability of 6 new integrated software products that run on the powerful M6809 microprocessor. Currently this software is available on the SWTPC 5-inch mini floppy FLEX disks (FLEX is a trademark of TSC). All programs are in relocatable format and may be loaded anywhere in memory. Also all programs do not require a special ROM since they contain their own I/O drivers.

The first program is a relocatable assembler which is about 12K bytes in length. It supports such things as common blocks (analogous to those in Fortran), local labels, M6800 & M6801 instructions, alphabetized or nonalphabetized cross reference table, and many other features too numerous to mention here. There is also the capability to assemble programs in absolute mode if desired. All source files are TSC compatable. There are a total of 53 error messages.

The second program is a relocating linking loader. This program will link up all required object modules and place the final program anywhere in memory. You also have the capability to relocate your common blocks independent of where your program was loaded.

The third program is a one pass link editor. This does essentially what the linking loader does except that the final object code is saved back on the disk as one object module instead of being loaded into memory. This object module may then be loaded anywhere in memory by the linking loader. Loading should be very fast since only one object module is loaded. This one pass link editor allows you to build up a library of relocatable 'Quick-load' modules.

The fourth program is a two pass link editor. This does essentially what the one pass link editor does except that it does it in 2 passes. The final object module generated by the 2 pass link editor is a little smaller than that generated by the 1 pass link editor since some optimization may be performed.

The fifth program is a global cross reference generator. This program allows you to tell at a glance all object modules which reference a particular internal label.

The last program is the object displayer. This outputs the header of any specified object module. Parameters in the header that are outputed are such things as program name, internal labels, external labels, time and date, and more.

The prices are as follows:

M69ASMB-1A-10, M69LOAD-1A*10	assembler and loader	$100.00
M69LNKA-1A-10	one pass link editor	$45.00
M69LNKB-1A-10	two pass link editor	$45.00
M69XREF-1A-10	global reference generator	$35.00
M69DISP-1A-10	object displayer	$25.00

All 6 programs and the users manual are available for $200.00. The users manual which describes all 6 programs may be purchased seperately for $15.00. This may be credited toward any future software purchase from us.

A catalog is available free from us for the asking.

Order from CINCITEX SOFTWARE
P.O. BOX 19365
CINCINNATI, OHIO 45219

COMPUTERWARE SOFTWARE SERVICES

CASH FLOW BOOKKEEPING SYSTEM

You don't have to wait for weeks until the accountant sorts out your receipts to see your expenses for last month - only to realize the information is too late to help you now! By merely entering your checking account information you can now have always-current visibility over your income/expense ledger.

Cash Flow Bookkeeping is a cash-based single entry bookkeeping system which allows the user to define multiple income and expense accounts. Deposits are assigned to income accounts while cash disbursements by check are assigned to expense accounts. Multiple expense assignments may be made for a single check, allowing easy recording of petty cash, credit card payments, etc. A chart of accounts code may be assigned to group various expenses into categories.

Year-to-date totals are maintained for every account, but may be zeroed at any time. All activity entered into the system is recorded and applied against the checkb k balance and the appropriate account. The systems automatically maintains the checkbook balance and will reconcile upon request. Means for recording automatic payments and manual adjustments is also provided.

Extensive reporting capabilities offer the following reports: Detail Account, Summary Account, Year-to-date Account, Written Checks, Outstanding Checks, Adjustments, and Deposits. Several sort sequences are provided and ranges within the sequences allow selective reporting. Subtotals are provided where applicable and grand totals are always printed.

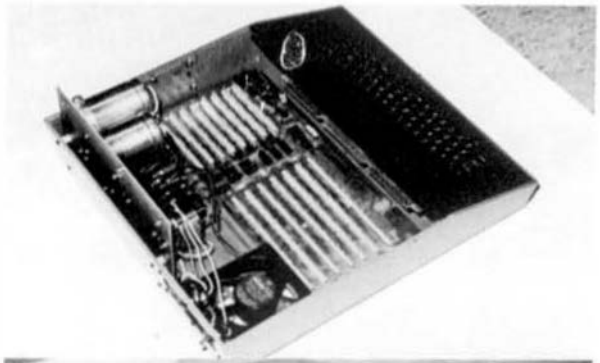
The Cash Flow Bookkeeping System can be interfaced with Computerware's Accounts Receivable and Accounts Payable Systems for a complete General bookkeeping system. Cash Flow Bookkeeping is immediately available for $149. The system is designed to run on a 40K 6800/6809 computer with a minimum of a dual 5" disk system. SSB DOS and Computerware Random BASIC are required. Versions for other disk operating systems are pending customer demand.

A manual describing the operation & reports of either system is available for $15.00 from COMPUTERWARE - 1512 Encinitas Blvd. - Box 66B - Encinitas, Calif. 92024 - (714) 436-3512 or 436-0282.

ECB-50

NEW SS-50 BUSS COMPUTER BASE KIT

The ECB-50 is an expandable support base for an SS-50 Buss microcomputer. The kit consists of a housing, power supply, motherboard, keyboard, and complete assembly instructions. The case is furnished ready for finishing. The power supply provides sufficient power to drive a full case of boards and two 5 1/4" floppies. The motherboard is a tinned, double sided pc card with full decoding of I/O ports and sockets for all IC's.



The keyboard has 67 keys, provides full ascii set, 5 user defineable keys, 4 mode operation, and requires single 5 volt supply.

The kit sells for $595.00 and is available from the "febe group inc.", 51 Hamilton Ave., York, PA 17404, (717) 854-0481.

Dealer inquiries invited.

SONEX SYSTEMS

BOX 238
WILLIAMSVILLE, NY 14221
716—634-9466

September 27, 1980

Mr. Dan Williams
68 Micro Journal
3018 Hamill Rd.
P.O. Box 849
Hixson, Tennessee 37343

Dear Dan,

I wanted to reply to you letter about the new TRS color computer. Like everyone in the industry I am going to watch it's development rather closely. I think with some modification a version of STYLOGRAPH can be made to work on it. While it may fall short in many respects as a text processing system I think many users will make do. This has certainly been the case with the old TRS computers. I think that you would be doing both yourself and your readers a disservice not to support this new computer. At the price I think that there are many of your readers, myself included, that are going to buy one just to play around with. I know that there are already people across the country that are tearing the thing apart to see how it works. You might consider running some articles on the hardware itself and possible interfacing to S-50 systems.

I am sure there will be some pressure to keep the 68 Micro Journal "pure". While I have some sympathy for this viewpoint I think in the long run all of us will benefit from the introduction of this new computer. In the first place, many people will be introduced to computers by the color computer and will eventually want to move up to a more powerful system. For these people the 68-50 series is the logical choice because of software compatibility. Also I think that the TRS computer gives all of us greater visibility in the market place.

Since the review of STYLOGRAPH appeared in your journal, a number of readers have asked about the version for the CIMIX 24x80 video board. This version will work directly on CIMIX systems. For users of SWTPC systems SONEX can supply a "primitive" monitor called SONEXBUG which will allow the CIMIX video board and a keyboard to be used as the console terminal. To take advantage of this system the user must first have a version of FLEX up and going with a normal serial terminal so that the appropriate patches (explained in the manual) can be made to FLEX. The monitor requires that the user have a SWTPC system which has 16 addresses per port and the space from $E000 to $EFFF fully decoded. SONEXBUG is available for $50.00.

Also, a number of readers have asked us about future enhancements. The next version of STYLOGRAPH will contain a few additional formatting and editing commands but the major change will be the addition of true proportional spacing for those users with NEC, Diablo, Qume or Centronics 737 printers. This will allow "typeset" printing on these printers. Registered STYLOGRAPH owners will be notified and can purchase the update for only $30.00. Another upcoming development is a compatible mail-list merge program that can take STYLOGRAPH files and do mail-list generation. It will also allow appending a number of STYLOGRAPH files so that continuous long documents can be produced from a series of shorter files. No release date for these developments are promis d but we hope to have them ready by early spring.

Keep up the good work!

Sincerely,

Bob

Robert S. Bundy

Support Our ADVERTISERS!

2502 Regal Oaks Lane

Lutz, Fla 33549

16 Oct 1980

Don Williams:

After you ran the listing of the source for SBUG-E in the June 1980 issue, I received a number of letters and calls concerning the 6809 CPU card and computer system. Therefore, to simplify things, I thought some reply comments to be in order.

Having re-written and expanded the monitor to 4K to provided many additional features I soon discovered that when making changes to the monitor it is necessary to preserve the restart vector entry point into the monitor at an address of $FFxx. The reason for this is that during power up the contents of the DAT RAM; IC-11 on the MP-09 CPU is indeterminate. Therefore IC-11 is disabled by $FFxx into IC5 which causes pin 2 of IC-11 to go high. With IC-11 outputs in a Tri-State high impedance state; R9-R14 pull address lines A12-A15 high to provide the most significant address nibble in lieu of IC-11's indeterminate data.

A second pitfall to avoid is to be CERTAIN that your system has pull up resistors on IRQ and FIRQ. This caution is intended for the user running an MP-09 in other than a SWTPCo mainframe. It has been mentioned before; but some continue to ignore the warning. If these lines are left to float; unreliable operation will result.

While I'm on the subject of the 6809 Monitor I might as well mention a related item. To modify the monitor required an E-PROM Programmer for which I purchased one of which I am pleased. The programmer I have is a Model EP-2A-79 from Optimal Technology Inc. The programmer is reasonably priced; well constructed; and promptly delivered. However; what I especially like is the fact that it uses personality modules and allows software to have nearly total control of it's operation. To some it may be a drawback that his/her software must control most everything; but it allowed me to write a control program that makes the programmer very friendly and powerful while outperforming some that cost several times as much. The program is approx. 4K + in length and is for 6809 5" FLEX. Written in assembly language; it is fully re-locatable and runs very fast. Capability is provided to save and load files to disk; examine and automatically verify ROM contents; program any number of bytes; and list ROM contents. The control port for the programmer is also selectable from the user keyboard at runtime allowing operation from any port. Being menu driven; an instruction manual is not required. You may recall that I sent you an early version of this program and the 4k monitor on diskette. Now that a final version is avialable I would consider providing a copy for publication if you so desire; though the heavilly commented source listings run between 30 and 50 pages each. Credit for contribution to the above also goes to W. Watason.

Sincerely;

Allen Clark

Editor's Note: Anyone interested in the expanded monitor and associated software, please let us know (68 Micro Journal). If enough interest is indicated I will try to publish the software talked about in Allen's letter above.

DMW

ESTHER
An Exercise in Artificial Intelligence
By Dale L. Puckett

ESTHER is Eliza plus. Artificial intelligence in pure 6800 code. Her source shows you how. Her object will amaze your friends.

ESTHER: remembers names, drops them, uses the player's name, answers third person replies and even echos keywords. Esther identifies more than 75 keywords and uses almost fifty sets of replys. A few of the sets contain as many as 21 replys to help her avoid redundancy. ESTHER features auto line length and runs in FLEXtm. She obeys TTYSET. She is both educational and fun.

ESTHER, written by 68 Micro Journal Contributing Editor, Dale L. Puckett, is the result of a two year long experiment with artificial intelligence in 6800 assembly language programming.

ESTHER is well structured and uses a number of common subroutines which makes it easy for experimenters to customize and do their own thing.

ESTHER'S documentation gives the table formats which makes it possible for the user to easily add or change keywords or replies.

ESTHER randomly inserts the players name in the conversation. Occasionally, she uses part of the player's reply in the middle of her answer or next question.

ESTHER identifies proper nouns and uses them in her replies. She also saves them for later use.

ESTHER has the ability to echo keywords. This allows her to respond to replys from the player which are in the third person.

The package comes with more than 2000 lines of well documented source code which can provide a base for a user's own experiments with artificial intelligence.

ESTHER will not allow the player to repeat an answer or avoid answering a question. And, she conjugates parts of the player's answer and uses the information in her reply or question.

Esther is available June 15, 1980. She will run in any 16 SS-50 bus computer which uses the FLEX 2.0 operating system. The introductory price is $39.95.

ESTHER is available from:

FRANK HOGG DENTAL LABORATORY
130 Midtown Plaza
700 W. Water Street
Syracuse, NY 13210

HEMENWAY ASSOCIATES, INCORPORATED

101 Tremont Street

Boston, Massachusetts 02108

617-426-1931

FOR IMMEDIATE RELEASE

M68000 Software

Hemenway Associates is pleased to announce two new utility software packages for the M68000 family of microprocessors. Both are designed to run under Hemenway Associates' powerful HA-SP/68000 operating system in a 32K byte RAM configured system.

The RA68000ML Resident Assembler and the LINK68000 Linking Loader (part of the assembler package) comprise a two-pass Macro Assembler and a one-pass Linking Loader. The AA68000 has full macro facilities and conditional assembly of up to 8 nested levels. It produces a listing, a sorted symbol table and generates relocatable and linkable object code.

It's fast--using a hash-coded symbol table and binary search of the mnemonic table. And it's sophisticated--allowing seperately assembled routines to share data, a feature which can be used to produce ROMable code.

All Motorola-defined opcodes are recognized, and a set of pseudo-op instructions which simplifies programming.

EDIT68000 is a resident line-oriented context editor which provides for the creation and editing of program and data files. It contains an extensive set of editing and file manipulation commands.

RA68000ML with LINK68000 ($350.00 U.S.) and EDIT68000 ($150.00 U.S.) are available off the shelf and include a user manual and object code on floppy disk.

M68000 FLOATING-POINT PACKAGE
($99.95)

This package provides a unified set of floating-point support subroutines for any 68000 microcomputer. Using a 32-bit binary floating-point format, these subroutines provide fast arithmetic operations, integer-float and float-integer conversions, float-ASCII string representation, and ASCII string-float conversions.

The floating-point format produces between 6 and 7 digits of precision and an exponent range from 1.E-20 to 1.E+20.

The package is designed to be ROM-able and all code is position independent. Each subroutine is re-entrant and interruptable. The package requires 1400 bytes of memory and may use up to 50 bytes of stack RAM.

Included in the package are the subroutines:

FADD.....addition
FSUB.....subtraction

FMUL.....multiplication
FDIV.....division
FIX.....float to 32-bit integer
FLOAT....32-bit integer to float
FTOA.....float to ASCII string
ATOF.....ASCII string to float

These subroutines are FAST---on a 68000 with a 8.0 MHz clock all the arithmetic subroutines run in 150 to 500 microseconds. A standardized calling sequence using registers A4 and A5 as pointers makes interfacing to these subroutines simple. Subroutine ATOF accepts free-format inputs, with optional number sign, decimal point, variable number of mantissa digits, optional one or two-digit signed exponent, etc. Also... the assembled source-code of the entire package is included.

- Fast execution
- Position-independent, ROM-able code
- Re-entrant subroutines
- Integer-float, float-integer
- ASCII-float with free-format input
- Float-ASCII string
- Standardized calling sequences

SOFTWARE ENGINEERING & TECHNICAL WRITING SPECIALISTS

FOR IMMEDIATE RELEASE

Ed Smith's Software Works, formerly of Redondo Beach, California has recently been acquired by Great Plains Computer Company, Inc. of Idaho Falls, Idaho. Prior to his untimely death in the Spring of this year, Ed Smith was producing some excellent development software for the 68XX world. His products included Relocating assemblers, Macro assemblers, Cross assemblers, relocating Disassembler-Source Generators, Monitor programs, and program Debugging tools. All of the products previously advertised by Ed Smith, are now available from Great Plains Computer Company. New products will be added to the line and all existing programs will improved and enhanced. For further information, write or call:

Great Plains Computer Company, Inc.
P.O. Box 916
Idaho Falls, Idaho 83401
208-529-3210

P & D Associates
1210 TODD ROAD
NEW PLYMOUTH OHIO 45654

Tel. 614 - 592 - 5721

September 8, 1980

NEW PRODUCT ANNOUNCEMENT

The TimeMinder

P & D is proud to announce the availability of a new clock/calender board for $30 systems. It is a 30 pin I/O card based on the OKI MSM5382 timekeeping chip. On board battery back-up and recharging circuit lets your computer always know the time (hour, minute, second - 12 or 24 hour format), day of week, day of month, month, and year. A leap year register takes care of February 29th when necessary.

The clock chip communicates thru an on-board 6821 PIA so that all setting and reading is done by software. Five extra PIA lines are brought to a Molex connector on the top edge of the board for use in control and measuring applications. One PIA line is used to control an on board "solid state buzzer" circuit for use in alarm applications.

With the proper software, many applications are possible. Such as: stopwatch; darkroom timer with audible alarm; event counting with time measurement; controlling home heating/cooling systems, security systems, etc. None of these applications would interfere with the board's primary job of time and date keeping. The board can also generate interrupts to the system at the rates of: 1024 per second; once per second; once per minute; once per hour.

The board comes with assembly and checkout instructions and demonstration software in the form of an assembly listing. The software is liberally commented and arranged so that it can easily be shortened for specific applications. Available separately is a 5 1/4 in floppy diskette for FLEX 2.0 (tm)* systems. It contains the demonstration software, several useful utilities for FLEX 2.0 *, and a patch for FLEX 2.0 * that causes it to load the date into FLEX and display the date and time instead of prompting for the date. Source files are included. Price of the bare board and documentation is $35.00. The diskette is $10.00. Include $2.50 s/h with each order.

We expect to have more application software available for this board in the near future. First up will be patches or USR routines for TSC's Basic.

* FLEX is a trademark of TSC.

Datricon
CORPORATION

7911 N.E. 33RD DRIVE
SUITE 200
PORTLAND OR 97211
503 284 8277

RESIDENT D-FORTH IN THE ACS-12 ALLOWS HIGH-LEVEL DEVELOPMENT WITH ONLY A POWER SUPPLY AND TERMINAL

FOR IMMEDIATE RELEASE

Applications Controller Series 12

The Applications Controller Series 12 is a single board computer featuring the Motorola 6800 microprocessor family. The ACS 12-PRO requires only a power supply and a terminal to operate.

The ACS 12 provides two programmable 16-bit timers, software controlled ACIA/MODEM interface on an RS232 serial port, two 8-bit parallel ports with handshake control (or one 16-bit parallel port with handshake control), and up to 4 kilobytes of static RAM and 6 kilobytes of Programmable Read Only Memory (PROM).

The on-board parallel port interfaces directly to an OPTO22 PB16 or equivalent while connected to a Peripheon 611 optical image digitizer.

The ACS 12-PRO is supplied with Datricon's 4K D-Forth software, a derivative of Forth. D-Forth is a task driven operating system that responds rapidly to all tasks upon demand. A 6k D-Forth features include a 6800 assembler, line editor, self-adjusting pointers for end-of-memory determination, and Automatic Baud rate search for the host terminal.

This board measures 4.5 inches by 9.6 inches by .5 inches. The STD BUS interface provides the user access to a large selection of input/Output devices available from Datricon or other manufacturers making it an excellent selection for process control or small computer users.

With 1K of static RAM, 4K D-Forth and manual, the ACS 12-PRO lists at $495 in Quantity 1, from stock.

The ACS 12-OEM lists at $395 in Quantity 1 from stock with volume discounts.

For further information call (503) 284-8277 or write:

Datricon Corporation
7911 N.E. 33rd Drive
Suite 200
Portland, Oregon 97211

ET-3400 USERS GROUP FORMING.

We are forming a group to collect and distribute helpful information for ET-3400 owners. We need articles, letters, programs, and etc.. Our first newsletter will be free to those submitting material. Please contact:

ET-3400 USERS GROUP
% CHARLES VAN DYKE
11231 Oak St.
El Monte, Ca. 91731
24 HOUR MESSAGE PHONE (213) 443-2237
CompuServe # 70250,463

THANK YOU

technical systems consultants, inc.

P.O. Box 2570 • 1208 Kent Avenue • West Lafayette, Indiana 47906 • (317) 463-2502

FLEX™ runs on Custom Hardware

A new version of the FLEX Disk Operating System is now available for users of custom or non-standard 6800 and 6809 systems. Developed by Technical Systems Consultants, Inc, it is fully compatible with versions of FLEX supplied by several manufacturers. FLEX has been in widespread use since early 1978 and supports such features as dynamic filespace allocation, random and sequential file accessing, user startup facility, user environment control, English error messages, and over 20 commands for all normal disk operations. This new version contains a 100-page "FLEX Adaptation Guide" which fully describes how the user can write his own disk and terminal I/O routines to adapt FLEX to most any hardware. The only major system requirement is a 256-byte, soft-sectored floppy disk. When the adaptation is complete, the user's system will be capable of running any standard FLEX software. The $150 list price includes the FLEX disk with editor and assembler, and a full set of manuals in a loose-leaf binder. Contact Technical Systems Consultants, Inc, PO Box 2570, West Lafayette, IN 47906; (317) 463-2502; Telex

F&D 5-8 Disk Controller

MDI-1 5&8 INCH DISK CONTROLLER

Available now for those who have acquired a spare disk drive or more, is the F and D Associates 5 and 8 inch disk controller board. Designed for 6800 or 6809 computers using a standard S50 bus system, the controller will handle either 5 or 8 inch floppy disk. It can be 'stuffed' with the WD1771 (single density) or WD1791 (double density) controller IC. Actually the WD1791 allows both single and double density formats. Also both 5 and 8 inch drives can be plugged onto the controller at the same. A total of 4 drives are supported.

Due to software considerations the 5 inch supports standard versions of FLEX; however, the 8 inch versions of FLEX will require some changes be made. At the present time only single density is supported for both the 5 and 8 inch drives. The manual states that double density updates (manual) will be available in the future.

The MDi-1 has a 'data-separator' (provided on board) if the WD1791 is installed. Users of the WD1771 must utilize the internal data separator, of the WD1771, or purchase the extra external data separator for $7.50 (bare board), which includes documentation and is available from F&D Associates.

Also detailed in the instructions is information stating that the use of 8 inch drives, with a CPU speed of 1.25 mhz, is possible, but the use of the separate data seperator is recommended for all 8 inch operations. This requires a resistor change on the 6800 SWTPC MP-A2 CPU card if the internal data separator is used and recommends a xtal change if the external separator is installed (to change CPU clock speeds). This was not tried on our test controller card but i would suggest that the external separator be used in all instances. It is a proven fact that the WD1771 has some problems 'grabbing' data when used with the internal data separator. Data acquisition rates will be enhanced if the external separator is installed properly. The price difference of $7.50 is probably well worth it.

CONFIGURATION

The disk controller is configured for different size disk by optional jumper pads on the board. Different controller chip options are jumper selected. Jumpers are provided for numerous other options on the board and the manual outlines their designation and application.

DOCUMENTATION

The manual is well done and indicates a continuing process of up-date. Complete construction information is contained as well as easy to read diagrams and board layout. Also included is software (disk boots) for both 5 and 8 inch drives.

Available as a separate package is a disk (5 or 8) for either SWTPC or SMOKE FLEX users ($20.00). This software contains programs to ease the transition to disk and this controller card.

It is suggested that FLEX be purchased from TSC or other AUTHORIZED sources and then configured to run with the MDi-1 controller board. Included in the software package is the following:

MDIDRV.BIN is a binary file that overlays the drivers of FLEX.

SETDRV.CMD sets the drive status, when called it displays the drive status and allows changes.

iNIT.CMD is a disk formatter. Unlike most formatters it also prompts for the date and includes it also on the disk.

LOADER is a multi-purpose program that prompts with B, L or G for the following functions. B) BOOT loads a program that has been linked into memory and reports the starting address. L) LOAD loads into memory a file by track and sector and reports the starting address. G) GO Jumps to the starting address reported by one of the other functions.

CONCLUSION

The board is well laid out but is slightly tight due to multiply options. it would be easy to develope solder 'bridges' due to close foils. Solder masking could have been a help in this case but it would have increased the price, which in some instances is the beauty of a board of this type (works well and has a minimal cost).

No silk screening but the manual has a very plain and easy to read outline and parts placement chart with drawings which should do the trick for even those with little construction experience.

One caution is in order. Some WD1771 chips are more critical that others. We have found by experience that some will work on one motherboard or controller board and not on others. This seems to be a timing problem, which varies from system to system and board to board. It is recommended that if you must purchase a 1771 or 1791, make certain you can

exchange the chip for another, if you are experiencing read errors or other marginal separator problems. The use of the external separator should eliminate this requirement but I can not say for sure as I do not have the separate data seperator. The price of the bare board with documentation is $35.00 and requires no 'tricky' or hard to find components.

Complete information can be secured from:

F&D ASSOCIATES

1210 Todd Road

New Plymouth, Ohio 45654

Dear Mr.Williams,

I have a homebrew 6800 system with a 68488 chip with Mikbug II and Audio Cassette Tape Driver in the Rom. Memory is 16kb dynamic and 4k static,the terminal is also homebrew using a standard t.v. set.
At the moment I am running an 8k SWTPC Basic on my system and everything works just fine.
Recently I acquired a Calcomp 142M Floppy Disc Drive (the one Motorola used in their Exorciser II unit) and a NEC uPD372 FDC chip.
I would like to know if there is anyone around who has used this chip in a 6800 system and if so if they would send me their schematics and maybe also their software for driving this chip. I will of course pay for all expenses made!
As for your magazine! It's so good to have a magazine dedicated to the 6800 and 6809 users,but I think too many articles are written about large systems with discs and 6800's,nothing noteworthy anymore about 6800 small systems. I think this is too bad,especially because a lot of people are still starting to enter the uP and 6800 market.
Who is going to give them a hand? I think your magazine should!
A note on what Mr.Paul Verburg had to say in the september '80 issue: Not only hardware is hard to come by in Holland,software is a problem too.

Sincerely yours,

W.Vleeshhouwer
Sim.Watinckstr.9'
2211 JM Noordwijkerhout
Netherlands

Alford and Associates
[illegible]
[illegible]

17 OCTOBER 1980 - NEW PRODUCT RELEASE - TREK-6864

Alford and Associates is pleased to announce TREK-6864. A completely new and exciting arcade-type game for users of memory-mapped displays and the 6800 processor.

TREK-6864 is the first comprehensive, multi-tasking, real-time game available for the 6800 processor. Although patterned after the many trek games available in Basic, TREK-6864 is a radical departure, as there is no question-answer dialogue after selecting the game difficulty option, until the game is won or lost! Doing nothing with TREK-6864 is the fastest way to lose the game!

The enemy ships will execute several battle plans in their effort to eliminate you. Energy will run out as you watch. Torpedoes will out-run you on the screen! Flashing messages, audible background accenting, and edge-of-the-chair action are youre with TREK-6864!

Requiring a 16x64 memory-mapped display, and a source of interrupts (such as an MP-T timer board), and optionally, an interface to a speaker for sound-effects, TREK-6864 will make solid detractors of the 6800 processor sit up and take notice!

TREK-6864 is available now for 16x64 memory-mapped displays, and is provided on disks running under TSC FLEX 1.0 & 2.0, SWTPCo mini-FLEX, and SSB DOS68. A SYSGEN program is provided on-disk to make custom adaptations easier. The program disk and manual are available now for only $24.95. Call or write:

ALFORD AND ASSOCIATES
P.O. BOX 6743
RICHMOND, VIRGINIA, 23230
804-320-6722

MR. GEOFFREY A. GASS
5240 S. W. DOSCH ROAD
PORTLAND, OREGON 97201

RE: HARDWARE HICCUP
'68' MICRO JOURNAL

DEAR MR. GASS:

BLESS YOUR SOUL FOR SENDING THE ABOVE ARTICLE TO '68' MICRO JOURNAL. I HAVE BEEN HAVING EXTREME PROBLEMS WITH MY A/2 BOARD FOR SOMETIME, EVEN TO THE EXTENT OF USING MY "A" BOARD AS A SEMI-PERMANENT REPLACEMENT, UNTIL I READ YOUR ARTICLE. WITHIN 10 MINUTES, YOUR DIAGNOSIS PAID OFF FOR ME (POWER-ON REQUIRED SEVERAL MANUAL RESETS). I HAVE CORRECTED THE PROBLEM BY USING SWTBUG AND MY A/2 WORKS LIKE IT SHOULD. IN THE MEANTIME, I'VE PUT ALL PROM ON A SEPARATE BOARD, SO NOW I HAVE TWO ALMOST IDENTICAL WORKING MPU'S, THE "A" AND THE A/2. THIS MAKES TROUBLESHOOTING A BREEZE WHEN USING A KNOWN GOOD PROCESSOR BOARD.

THANKS AGAIN FOR YOUR ANALYSIS OF THIS PROBLEM AND ITS SOLUTION.

SINCERELY,

ERNEST STEVE WATSON

CC: '68' MICRO JOURNAL

U.S. POSTAL SERVICE
STATEMENT OF OWNERSHIP, MANAGEMENT AND CIRCULATION

1. TITLE OF PUBLICATION: 68 MICRO JOURNAL
2. PUBLICATION NO.: 4 6 6 5 1 0
3. DATE OF FILING: 10/9/80
4. FREQUENCY OF ISSUE: Monthly
A. NO. OF ISSUES PUBLISHED ANNUALLY: 12
B. ANNUAL SUBSCRIPTION PRICE: $18.50
5. LOCATION OF KNOWN OFFICE OF PUBLICATION: 5131 Ar[illegible] Blvd., Chattanooga, TN 37421
6. LOCATION OF THE HEADQUARTERS OR GENERAL BUSINESS OFFICES OF THE PUBLISHERS: 3018 Hamill Rd., Hixson, TN 37343
7. NAMES AND COMPLETE ADDRESSES OF PUBLISHER, EDITOR, AND MANAGING EDITOR
PUBLISHER (Name and Address): Donald M. Williams Sr., 3018 Hamill Rd., Hixson, TN 37343
EDITOR (Name and Address): Same
MANAGING EDITOR (Name and Address): Larry E. Williams, 3018 Hamill Rd., Hixson, TN 37343
8. OWNER

NAME	ADDRESS
Computer Publishing Inc.	3018 Hamill Rd., Hixson, TN 37343
Donald M., Joyce F., Larry E., Mary E., Thomas E., Williams	same

9. KNOWN BONDHOLDERS, MORTGAGEES, AND OTHER SECURITY HOLDERS OWNING OR HOLDING 1 PERCENT OR MORE OF TOTAL AMOUNT OF BONDS, MORTGAGES OR OTHER SECURITIES

NAME	ADDRESS
None	

10. FOR COMPLETION BY NONPROFIT ORGANIZATIONS AUTHORIZED TO MAIL AT SPECIAL RATES

11. EXTENT AND NATURE OF CIRCULATION	AVERAGE NO. COPIES EACH ISSUE DURING PRECEDING 12 MONTHS	ACTUAL NO. COPIES OF SINGLE ISSUE PUBLISHED NEAREST TO FILING DATE
A. TOTAL NO. COPIES PRINTED (Net Press Run)	6851	7209
B. PAID CIRCULATION 1. SALES THROUGH DEALERS AND CARRIERS, STREET VENDORS AND COUNTER SALES	3511	3409
2. MAIL SUBSCRIPTIONS	2987	3411
C. TOTAL PAID CIRCULATION	6498	6820
D. FREE DISTRIBUTION BY MAIL, CARRIER OR OTHER MEANS SAMPLES, COMPLIMENTARY, AND OTHER FREE COPIES	163	184
E. TOTAL DISTRIBUTION	6661	7004
F. COPIES NOT DISTRIBUTED 1. OFFICE USE, LEFT OVER, UNACCOUNTED, SPOILED AFTER PRINTING	190	205
2. RETURNS FROM NEWS AGENTS	0	0
G. TOTAL	6851	7209

12. I certify that the statements made by me above are correct and complete.
SIGNATURE AND TITLE OF EDITOR, PUBLISHER, BUSINESS MANAGER, OR OWNER
Donald M. Williams Sr.

PS Form 3526 (Page 1) (See instructions on reverse)

CLASSIFIED

NO REASONABLE OFFER REFUSED. UNUSED ED SMITH SOFTWARE WORKS M-68RR-SS8R, RR, MAC, MINI DISK AND DOCUMENTATION. MARTIN POPPE (802)644-5196.

***

FOR SALE:ONE SWTPC CT-82 $750.00 CALL (615)870-9869 ASK FOR TOM WILLIAMS.

***

CT-64 $150.00, CRT MONITOR $50.00, PERCOM LFO- 00 AND SOFTWARE $175.00. SWTPC 6800, 28K, FLOPPY CONTROLLER, FLEX 2, TEXT PROCESSOR $600.00. TIM TIBBOT, 1812F WOODMAR DR, HOUGHTON, MI 49931 (906)482-8806.

***

LEAR SEIGLER ADM-3 CRT FOR SALE. ANY REASONABLE OFFER ACCEPTED. JIM PAZARENA, BOX 550 QUEEN CHARLOTTE, BC CANADA VOT1SO

***

WANTED:SWPTC 16K DYNAMIC RAM BOARD, MP-A CPU BOARD, MP-P POWER SUPPLY. PETE STARK, PO BOX 209, MT. KISCO NY 105 9.

***

PERFECT GIMIX 16K STATIC 2MHZ SS-50 MEMORY BOARDS $250.00. (PAID $298). SWARTZ, BOX 1384, IOWA CITY, IA 52244 (800)553-6235 EVENINGS.

***

FOR SALE:GT-6144, MP-N, MP-T, (2) 4K BOARDS, CT-64 AND MONITOR. BEN BEZENEK (612)733-3650.

***

FOR SALE:SMOKE SIGNAL BROADCASTING DISK CONTROLLER CARD, WITH THE FOLLOWING SOFTWARE ON 8" FLOPPY DISK, TEXT EDITING, DISK FILE BASIC AND MNEMONIC ASSEMBLER. $175.00 PAUL LAMAR (213)374-1673.

***

SWPTC 6800-2, AC-30, TC-3, PMB-1, CT-64, GT-6144, PPG-J, TSC SOFTWARE, EXTRAS. INVESTED $2400, ASKING $1400. J QUINLAN 5145 NORTH 10TH, ARLINGTON, VA. 22205, (703)243-4748.

***

FOR SALE: SWPTC 68/2 COMPUTER SYSTEM WHICH CONSISTS OF 32K RAM 2 SERIAL, 1 CONTROL AND 1 PARALLEL INTERFACE. MF-68 MINI WITH SA-400'S AND I TERFACE. 1 DECWRITER II WITH NUMERIC KEYS, 20 MA AND RS-232 INTERFACE. 1 CAT MODEM AND SYSTEM SOFTWARE PLUS ALL DOCUMENTATION FOR ALL HARDWARE LISTED. THIS SYSTEM IS STILL IN USE TODAY AND HAS BEEN KEPT CLEAN AND IN GOOD WORKING ORDER! SOFTWARE INCLUDES MINI-FLEX, SWTPC BASIC, TSC BASIC, TSC EDITOR, TSC TEXT PROCESSOR, TSC ASSEMBLER, TSC DEBUG, COM-PUTERWARE'S LEARN BASIC (DISK) AND ALL THE TSC UTILITY'S, PLUS INCLUDED IS A CT-1024 AND GT-6144. ALL ITEMS LISTED FOR ONLY $3000.00. SORRY, BUT I CAN ONLY SELL THIS AS A SYSTEM. P.S. ALL SOFTWARE HAS IT'S OWN DOCUMENTATION. *ORIGINAL DOS AS SUPPLIED WITH MF-68. CALL (201)689-2634 BETWEEN 9AM TO 5PM, OR (201)852-3954 BETWEEN 7PM TO 10PM, ASK FOR, "DALLAS".

HELP

DO YOU KNOW WHERE A PERSON COULD PICK UP A MEK 6800 D2 KIT, MEK 6802 D3 KIT, OR A MEK 68R2, CRT INTERFACE. P.S.-WHAT PRICE RANGE? N. WRIGHT, 7960 S. WILDWOOD DR., OAK CREEK, WI 53154.

SMOKE SIGNAL BROADCASTING

Presents

3 Powerful New SS-50/SS-50C Boards

DCB-4
Disk Master
Double Density Controller Board and DOS68D Double Density DOS
$449.00

The new DCB-4 is a truly state-of-the-art development which allows up to 366K bytes to be stored on a single 5¼" disk and has these outstanding features:

- Up to four 5¼"and four 8" drives can be handled in the same system with a user definable logical unit table. (DOS68D will be compatible with future hard disk systems).
- Under software control, the user can select the following for any drive:
  - ☆ Single sided or double sided operation.
  - ☆ Single density or double density data.
  - ☆ 5¼" or 8".
  - ☆ Stepping Rate.
  - ☆ 40 track or 35 track density on double sided 5¼" drives.
  - ☆ User can select the system boot configuration.
- Occupies only 16 bytes of memory space (F760-F76F standard). User selectable to any 16 byte address space.
- Can read and write a single sector by itself. On-board buffer memory allows full interrupt capability in interrupt driven systems. Once data transfer has been initiated, no more processor time is required.
- Contains extended decoding circuitry for extended addressing per SS–50C bus which can be enabled by an option jumper.
- SSB provides a means for copying software written by older versions of DOS68 to be read by DOS68D. All new media formatted by DOS68D can be read by all older versions of DOS68. DOS68 is SSB's 6800 disk operating system.
- Track 0 of side 0 is recorded in single density per IBM standard.
- Phase-locked-loop assures highest data integrity attainable.

All of these features are available for immediate delivery on one standard 5½" x 9" 50 pin SS–50/SS–50C card for only $449.00. The price includes DOS68D version 5.1, MONITOR object code on diskette, and a manual with the source listing.

SCB-69
Super Computer Board
6809CPU Board
$299.00

The most versatile 6809 CPU Board on the market is now available from Smoke Signal Broadcasting and has the following features:

- Standard 2 MHz operation.
  (Shipping 1.5 MHz until 68B09 available)
- 20 bit address generation for up to 1 Mbyte of memory. Uses an improved address translation RAM which is compatible with present extended addressing schemes yet requires much less overhead when used in multi-user systems.
- All on-board devices can be switch selected to occupy any or all extended pages. Any on-board device may be disabled and its memory space is then available for external memory.
- Standard real–time clock (time–of–day, day–of–week, day–of–month) with battery back up capable of generating programmable interrupts.
- Up to 20K of EPROM can be installed on the CPU Board.
- Standard 1K of RAM on board.
- Includes improved 6809 Monitor (and source listing).
- Contains an FPLA for decoding EPROM address and optional devices. Switches are used to select 2K/4K EPROM and Fast/Slow I/O.
- Contains provision for optional 9511/9512 floating point processor.
- NMI line is user selectable to work with either SS–50 or SS–50C busses.

Price for the new SCB–69 is only $299.00 for an assembled, burned-in fully tested board.

M-32-X
32K
Memory Board
$539.00 $439.00

The first and only 32K Static Ram Board on standard size (5½" x 9") SS–50/SS–50C Bus Circuit Card is made by Smoke Signal.

- Switch selectable to any 4K boundary.
- Any 4K block may be switch enabled or disabled.
- Fully compatible with SS–50C extended addressing (allows memory decoding up to 1 Mbyte).
- Extended addressing capability may be switched off for compatibility with SS–50 systems.
- Gold Bus Connectors for high reliability.
- Guaranteed 2MHz operation (tested at 2.2 MHz).
- Low power consumption – 8 volts at 2.4 amps typical.

M–32–X 32K Memory Board is priced at $539.00.

M–24–X 24K Memory Board expandable to 32K, is $439.00.

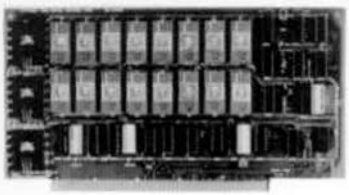
And our M–16–X 16K board is back to the old price of $299.00.

SMOKE SIGNAL BROADCASTING®

31336 Via Colinas, Westlake Village, CA 91361, (213) 889-9340

DIGITAL RESEARCH COMPUTERS
(214) 271-3538

32K S-100 EPROM CARD
NEW!



$74.95 KIT

USES 2716's

Blank PC Board - $34

ASSEMBLED & TESTED ADD $30

SPECIAL: 2716 EPROM's (450 NS) Are $14.95 EA. With Above Kit.

KIT FEATURES:
1. Uses +5V only 2716 (2Kx8) EPROM's
2. Allows up to 32K of software on line!
3. IEEE S-100 Compatible.
4. Addressable as two independent 16K blocks.
5. Cromemco extended or Northstar bank select.
6. On board wait state circuitry if needed.
7. Any or all EPROM locations can be disabled.
8. Double sided PC board, solder-masked, silk-screened.
9. Gold plated contact fingers.
10. Unselected EPROM's automatically powered down for low power.
11. Fully buffered and bypassed.
12. Easy and quick to assemble.

16K DYNAMIC RAM PARTIALS
LOOK! **INTEL 2108 8K X 1 RAMS** LOOK!
8 FOR $9.95 32 FOR $35
FACTORY PRIME!

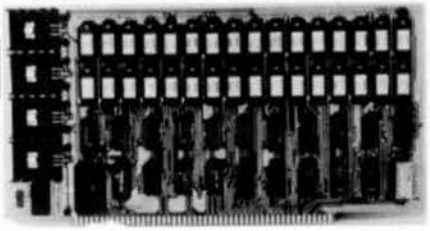
Huge special purchase of INTEL Dynamic RAM's. These are 2108-4, 300NS, 8K, Ceramic DIP. The 2108 is the INTEL 2116 (16K) tested for either upper or lower 8K only. These are factory prime. Full Spec. See INTEL 1978 Cat. for details or Memory Design Handbook for application data. Both IMSAI and EXTENSYS did mfg. S-100 RAM boards using these devices. — P.S. These devices will not work in the SD EPANDORAM™. Please specify upper or lower 8K. (S1626 or S1627). A super easy RAM to interface to a Z80, 16 PIN DIP.

FOR 4MHZ — PRICE CUT! **LOW POWER - 300NS 2114 RAM SALE!** 8 FOR $37.50

4K STATIC RAMS, MAJOR BRAND, NEW PARTS.
These are the most sought after 2114's, LOW POWER and 300NS FAST.
8 FOR $37.50

16K STATIC RAM KIT-S 100 BUSS

PRICE CUT!
$199.95 KIT
FOR 4MHZ ADD $10



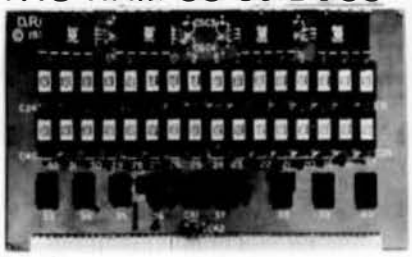
KIT FEATURES:
1. Addressable as four separate 4K Blocks.
2. ON BOARD BANK SELECT circuitry. (Cromemco Standard!) Allows up to 512K on line!
3. Uses 2114 (450NS) 4K S tatic Rams.
4. ON BOARD SELECTABL E WAIT STATES.
5. Double sided PC Board, with solder mask and silk screened layout. Gold plated contact fingers.
6. All address and data lines fully buffered.
7. Kit includes ALL parts and sockets.
8. PHANTOM is pampered to PIN 87.
9. LOW POWER: under 1.5 amps TYPICAL from the +8 Volt Buss.
10. Blank PC Board can be populated as any multiple of 4K.

BLANK PC BOARD W/DATA-$33
LOW PROFILE SOCKET SET-$12
SUPPORT IC'S & CAPS-$19.95
ASSEMBLED & TESTED-ADD $35

OUR #1 SELLING RAM BOARD!

16K STATIC RAM SS-50 BUSS

PRICE CUT!
$210 KIT
FULLY STATIC!
FOR 2MHZ ADD $10



FOR SWTPC 6800 BUSS!

ASSEMBLED AND TESTED - $35

KIT FEATURES:
1. Addressable on 16K Boundaries
2. Uses 2114 Static Ram
3. Fully Bypassed
4. Double sided PC Board. Solder mask and silk screened layout.
5. All Parts and Sockets included.
6. Low Power: Under 1.5 Amps Typical

BLANK PC BOARD—$30 COMPLETE SOCKET SET—$12
SUPPORT IC'S AND CAPS—$19.95

NEW! **STEREO! S-100 SOUND COMPUTER BOARD** NEW!

At last, an S-100 Board that unleashes the full power of two unbelievable General Instruments AY3-8910 NMOS computer sound IC's. Allows you under total computer control to generate an infinite number of special sound effects for games or any other program. Sounds can be edited in BASIC, ASSEMBLY LANGUAGE, etc.

KIT FEATURES:
- TWO GI SOUND COMPUTER IC'S
- FOUR PARALLEL I/O PORTS ON BOARD
- USES ON BOARD AUDIO AMPS OR YOUR STEREO
- ON BOARD PROTO TYPING AREA
- ALL SOCKETS, PARTS AND HARDWARE ARE INCLUDED
- PC BOARD IS SOLDERMASKED, SILK SCREENED, WITH GOLD CONTACTS
- EASY, QUICK, AND FUN TO BUILD. WITH FULL INSTRUCTIONS.
- USES PROGRAMMED I/O FOR MAXIMUM SYSTEM FLEXIBILITY

Both Basic and Assembly Language Programming examples are included.

COMPLETE KIT! **$84.95** (WITH DATA MANUAL)

BLANK PC BOARD W/OATA $31

SOFTWARE:
SCL™ is now available! Our Sound Command Language makes writing Sound Effects programs a SNAP! SCL™ also includes routines for Register-Examine-Modify, Memory-Examine-Modify, and Play-Memory. SCL™ is available on CP/M* compatible diskette of 2708 or 2716. Diskette - $24.95 2708 - $19.95 2716 - $29.95 Diskette includes the source. EPROM'S are ORO at EOOOH.

4K DYNAMIC RAM BLOWOUT!
SAME AS INTEL 2107B!
4K RAMS AT AN UNBELIEVABLE 50¢ EACH!!!

Prime, new, National Semi., 1979 date coded, full spec, parts, N.S. #MM5280-5N. Same as INTEL 2107B-4, T.I. TMS4060, NEC uPD411, etc. We bought a HUGE QTY. from a West Coast Distributor at truly DISTRESS PRICES! One of the most popular and reliable RAM's ever made. These parts have been used by almost all Major Computer Main Frame Mfg. the world over! Arranged as 4K x 1, 270 NS Access Time, 22 Pin Dip. These units DO NOT use multiplexed addressing, thus making REFRESH and other timing very simple. See INTEL MEMORY DESIGN HANDBOOK for full application notes. The NAT. SEMI. MEMORY DATA BOOK is available at most Radio Shack Stores. Prime units in original factory tubes!

(With Pin Out Data) **#5280-5N 4096 BITS x 1 270 NS ACCESS**
8 FOR $4.95 32 FOR $16
FACTORY CASE (450 PCS) — $180
Sockets Special: 22 Pin Low Profile (With Purchase of 5280's) 8 FOR $1.

COMPUTER PARTS SPECIALS

74LS175 - .99	8035 Intel Single Chip CPU - 5.95
74LS240 - 1.79	Signetic 2901 4 Bit Slice - 6.95
74LS241 - 1.79	AMD 2903 4 Bit Super Slice - 12.50
74LS244 - 1.79	AMD 29705 Dual Port RAM - 8.95
74LS373 - 1.99	

NEW! G.I. COMPUTER SOUND CHIP

AY3-8910. As featured in July, 1979 BYTE! A fantastically powerful Sound & Music Generator. Perfect for use with any 8 Bit MicroProcessor. Contains 3 Tone Channels, Noise Generator, 3 Channels of Amplitude Control, 16 bit Envelope Period Control, 2-8 Bit Parallel I/O, 3 D to A Converters, plus much more! All in one 40 pin DIP. Super easy interface to the S-100 or other Busses. **$11.95 PRICE CUT!**

SPECIAL OFFER: ~~$14.95~~ each Add $3 for 60 page Data Manual.

Digital Research Computers
(OF TEXAS)
P.O. BOX 401565 • GARLAND, TEXAS 75040 • (214) 271-3538

TERMS: Add $1.50 postage. We pay balance. Orders under $15 add 75¢ handling. No C.O.D. We accept Visa and MasterCharge. Tex. Res. add 5% Tax. Foreign orders (except Canada) add 20% P & H. 90 Day Money Back Guarantee on all items; Orders over $50, add 85¢ for insurance.

*TRADEMARK OF DIGITAL RESEARCH. NOT ASSOCIATED WITH DIGITAL RESEARCH OF CALIFORNIA, THE SUPPLIERS OF CPM SOFTWARE.

DATA BASE MANAGER

The Universal Data Research Data Base Management System (DBMS) is a comprehensive group of programs that allow a virtually untrained person to store and recall vast amounts of information in a computer system to meet individual requirements.

The DBMS is written in TSC Extended Disk Basic and requires at least 48K of memory to operate. All programs use a parameter file to allow easy adaption to individual systems.

The user is guided through these extensive programs by menus and sub-menus grouped by type of function. By simply answering prompts the user can create files, store any type of data and recall or manipulate it. The complex task of maintaining data files on the disk is completely taken care of by the programs, the size of the files is only limited by the disk storage capacity of the computer system.

Transparent to the fixed sector length, sub-records of related information are created only to the size required to conserve disk space. These sub-records can contain as many as 27 different fields of information. Each field in turn can contain either alphanumeric, integer or floating point data.

For those users who wish to write their own specific tasks for the database a complete source listing of all the subroutines is included at no extra charge.

BRIEF DESCRIPTION OF PROGRAM TYPES

CREATE DATA FILES The user specifies the file name, password and type of different data he wishes to store.

BUILD A FILE The user specifies the file name and is then prompted through the fields, he has previously specified, to enter the data he wishes to store. After verification the data is stored and the user prompted for the next group of data.

EDIT A FILE The user specifies which record he wishes and the data for that record is displayed. The user then has the option to alter any data contained by that record. Records can be specified by the actual record number or by the data being lookad for by the user.

SORT PROGRAMS To organize the data in the most meaningful order the user can sort any file by any field, create a sorted keyfile or merge two sorted files together.

REPORT PROGRAMS To meet the users individual needs reports can be completely customized. From a single record, labels or paginated sheet the user need only select the data he wishes to print and see only the data that meets a specified criteria. Report dfinitions can be saved and used to rerun new reports at any time.

FILE UTILITY PROGRAMS Enable the user to delete records, compress files, or modify any specified field data throughout the entire file. The user may also transfer data from one file to another.

GENERAL UTILITY PROGRAMS A group of utilities is provided to allow the user to view the directory of a disk, change the system date, print a source of a program, compare two programs for any differences or search a program for the occurence of any specified string without ever having to leave the DBMS.

ADVANTAGES OF THE UDR - DBMS

Learn to use the DBMS quickly. There is no programming language or set of commands to learn and remember.

Enjoy the time savings previously required to collect data and prepare it.

Get the report you want, in the exact format you want, at any time.

See your data sorted in its most meaningful order.

Create your files quickly and maintain them easily.

Keep your data current with easy editing and file manipulaion.

- - - DEALER INQUIRIES INVITED - - -

UNIVERSAL DATA RESEARCH INC.

FLEX	$ 350-00
UNIFLEX	$ 450-00

2457 WEHRLE DRIVE • BUFFALO, NEW YORK 14221 • 716-631-3011

THE MICRO WORKS

SEE US FOR ROBOT POWER!

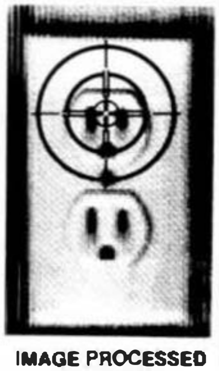


IMAGE PROCESSED BY DS-68A

In any decision-making process, the key factor is information. The Micro Works' line of video digitizers can provide your Droid with all the necessary visual information to make intelligent and accurate decisions about its environment.

Since our digitizers are software driven, your creation can extract only the information that it needs to solve a problem— from detecting ambient light conditions to successfully zeroing in on a power source (right).

If you need more information on the DS-68A, give us a call.

P.O. Box 1110, Del Mar, CA (714) 942-2400

SANSASKA SYSTEMS PRESENTS:

SBUG09: the BETTER monitor for the 6809 with these features in the base (A) module: examine/change memory in 3 formats, vectoring of all interrupts, search for additional modules, S1 format punch and read tape, BCB'S for I/O, search memory for a match, 8 transparent breakpoints, proceed from breakpoint, register display/change, JSR to program, GO (RTI) to program, 16 bit add and subtract, abort, random number generator, and mnemonic expansion capability.

The expansion module (B) adds these features: read tape in S1 format with offset, search memory for a 16 byte masked match, memory checksum, memory fill, copy memory from place to place, test RAM, exit to FLEX, and boot the disk.

The standard version is for use with FLEX09* and 5 1/2" minidisks. Other versions and customization are available on request. (If your disk boot is less than $40 bytes long I will substitute it at no additional charge.) Please specify your system configuration with your order.

MEDIA	PART A	PART B
MINIDISK	$30.00	$20.00
2708(2)*	$45.00	$25.00
2716(1)*	$45.00	$35.00

*Prices subject to change with EPROM prices.

STASM09: an overlay for the TSC* 6809 resident disk assembler, ASM09**. STASM09 allows the use of structured constructs in assembly language. STASM09 supports IF's, DO and COUNT loops with nesting up to 8 levels. STASM09 allows you to write code that looks like:

```
+ENTER
 LDA XYZ
 IF A,EQ,#5
 .DO
 ..DECB
 ..BREAKIF B,EQ,#45     EXIT THE DO LOOP WHEN (B)=45
 ..MORE CODE
 .ENDO
 ELSEF                  A DID NOT = 5
 .IF B,ON,#$F0          EXECUTE IF (B).AND.$F0 <> 0
 ..MORE CODE
 .ENDF
 ENDE
 RTS
```

STASM09 is available only on a 5 1/2" MINIDISK and includes the source and MACRO files necessary to run. It costs only $25.00.

*FLEX is a trademark of TSC
**ASM09 is a trademark of TSC

SANSASKA SYSTEMS
3311 CONCORD BLVD
CONCORD, CA 94519

1-415-676-2268
Evenings Pacific Time

BLITZ

SCREEN EDITOR FOR THE CT-82

- IDEAL FOR WORD PROCESSING OR PROGRAMMING
- THERE IS NO FASTER / EASIER WAY TO EDIT TEXT
- IT ALL HAPPENS IMMEDIATELY ON THE SCREEN SO YOU SEE EXACTLY WHAT YOU ARE DOING: INSERT CHARACTER, DELETE CHARACTER, INSERT LINE, DELETE LINE, SCROLL UP, SCROLL DOWN
- RUNS ON 6800 OR 6809 UNDER TSC's FLEX
- AVAILABLE ON 5 OR 8 INCH DISKETTE
- BEST OF ALL — YOU CAN BUY THE ENTIRE ASSEMBLY LANGUAGE SOURCE CODE, SO YOU CAN ADD YOUR OWN CUSTOM FEATURES
- FROM THE COMPANY THAT BROUGHT YOU THE MICROPI 4-USER PILOT/BASIC/EDITOR PACKAGE

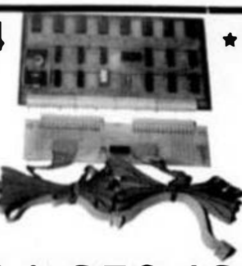
$50 — OBJECT ONLY

$100 — SOURCE AND OBJECT

MICROPI
2445 N. Nugent
Lummi Island, Washington 98262

208-734-8248

⋆ CT-64 ⋆ CT-1024



⋆ DMA VIDEO ADAPTER FOR YOUR TERMINAL

- **DMA** (ability to update anyplace on the screen directly)
- **HIGH SPEED DISPLAY** (fast as any video board)
- **KEYBOARD CONTROL** (of baud rate and paging /scrolling)
- **DOCUMENTATION** (includes source listing that replaces Outee)

J.B.I. adapter with memory $162.95, J.B.I. adapter without memory $149.95. Source Code on Disk $5.00 — Tape $3.50
Provide your system configuration and software.
Terms: cash, MC, Visa or C.O.D. plus $3.50 shipping and handling.

Johnson Micro Computer
2607 E. Charleston
Las Vegas, Nev. 89104
1-702-384-3354

HEMENWAY ASSOCIATES
SOFTWARE SOURCE BOOKS™

Your 6800 ia up and running with HEMENWAY ASSOCIATES' complete software system. Software Source Books provide a powerful yet extensible programming package for business, scientific, or personal uses.

Combining detailed descriptions with COMPLETE SOURCE CODE LISTINGS, these books explain the internal operations and algorithms used in HEMENWAY ASSOCIATES' popular systems software.

Imagine getting a complete 6800 software library, and at these suprisingly low prices.

Remember, these are not just books; they are Software Source Books™ complete software resources! Order them today; VISA and MASTERCHARGE accepted.

CP/68 OPERATING SYSTEM

The moat powerful operating system available tor the 6800 family of microprocessors, this disk-baaed system features great flexibility. The user can add commands for special purposes. A single transient Peripheral Interchange Program (PIP) transfers data between devices. The system is relocatable anywhere in memory and fits in leas than 8K. Other features include device-independent I/O and dynamic file allocation.

U.S. $34.95
Int. $52.45

XA6809
MACRO LINKING CROSS-ASSEMBLER

This new two-pass program generates relocatable and linkable code (requires LINK68). Reaident on any 6800 system, XA6809 lets you produce code for a 6809 right now. This assembler has full macro facilities and features a COMMON section for the production of ROMable code, Conditional Assembly and fast execution.

U.S. $24.95
Int. $31.50

LINK68 LINKING LOADER

This is a one-pass linking loader which allows separately translated relocatable object modules to be loaded and linked together to form a single executable load module, and to relocate modules in memory. It produces a load map and a load module in Motorola MIKBUG loader tormat. This book provides everything necessary tor learning about this syatem and the nature of linking loader design in general.

U.S. $7.95
Int. $11.95

RA6800ML RELOCATABLE
MACRO ASSEMBLER

This two-pass assembler produces a program listing, a sorted symbol table listing, and relocatable object code. The object code is loaded and linked with other assembled modules using LINK68. This book fully describes the 6800 assembly language and all major routines used, and includes flow charts, details on interfacing the assembler. Cross-referenced, showing all calling and called-by routines, pointers, flags and temporary variables.

U.S. $24.95
Int. $37.45

Structured BASIC language
STRUBAL+™COMPILER

The compiler features variable precision from 4 to 14 digits for business or scientific uses and Structured Programming forms. It produces Relocatable and linkable code. You can create data structures with mixed data types, COMMON and DUMMY sections. STRUBAL+™ includes a complete scientific package. It allows for string-handling and ia extensible.

U.S. $49.95
Int. $74.95

Please send the following goods:
_____ copies CP/68 OPERATING SYSTEM
_____ copies STRUBAL+™ COMPILER
_____ copies XA6809 CROSS ASSEMBLER
_____ copies RA6800ML MACRO ASSEMBLER
_____ copies LINK68 LINKING LOADER

HEMENWAY ASSOC., INC.
101 TREMONT STREET
BOSTON, MA 02108
(617)426-1931

FOR NORTH AMERICA: ADD $.75/BOOK POSTAGE & HANDLING or $1.50/BOOK FOR FIRST CLASS
ALL OTHER DESTINATIONS: ADD $2.00/BOOK POSTAGE & HANDLING or $3.50/BOOK FOR PRIORITY MAIL

Name _____ Title _____ Company _____

Street _____ City _____ State/Country _____ Postal Code _____

☐ CHECK ENCLOSED IN THE AMOUNT OF $
☐ BILL VISA ☐ BILL MASTERCHARGE
CARD NUMBER _____ EXP. DATE _____
☐ FIRST CLASS/PRIORITY ☐ BOOK RATE

THOMAS INSTRUMENTATION

*** SS-50 MODEM CARD ***

*Auto answer/originate *Uses the Bell 103 Modem standard (0-300 baud) *Use with DAA-CBS interface *Dial pulsing capability & software listings included for user "dial-up" and/or "answer" **Extra features: *Tone dial capability *2 extra RS-232 serial ports *Real time clock/calendar *2 extra parallel ports

*Asm. & tested, with extra features			$395.00
*Asm. & tested, without extra features			$325.00
*Special parts kit (bare board, 2 filters, relay, and transformer)			$195.00
*Software object and source on flex disk . . .	$10.00	*bare board	$ 49.00

A/T $425.00 ***** SS-50 24K RAM CARD ***** B/C $49.00

*Decoded for extended addressing *6800, 6802, 6809 compatible *6-4K blocks individually addressable 0-F *Low power consumption (typ. 3 amps) *Add memory 1K at a time using low cost 2114L's ($5.00 ea) *Gold edge connectors
A/T with 16K $325.00; with 8K $225.00; with 1K $150.00

A/T $195.00 ***** SP-1 ***** B/C $49.00

*Perfect for the user who wishes to design his own special interface, but doesn't want the bother of decoding and interfacing to the processor buss. Three PIA's, four ACIA's, and one PTM (3-6821, 4-6850, 1-6840) are already buffered and decoded. Three TO 220 regulators (+5, +12, -12) *Pad spacing permits the use of most standard sockets from 8 to 64 pins *Accommodates a mix of 38 14 & 16 pin sockets

A/T $95.00 ***** TRANSITION CARD ***** B/C $49.00

*A component part of Thomas Instrumentation's Backplane System *Contains all the necessary hardware to interface 8 SS-30 slots to one SS-50 slot *Contains all the logic necessary for use with a 6800 or 6809 system *Provisions for optional (not supplied) on-board baud rate generator and slow memory have been made

A/T $95.00 ***** CASSETTE INTERFACE CARD ***** B/C $49.00

*Plugs into Thomas Instrumentation's CPU Card *Uses circuitry licensed from and compatible with the JPC High Speed interface *Includes dual relays for motor control
Software drivers on cassette $25.00 Software drivers in EPROM $35.00

*RCA Keyboards Model 601 $80.00 Model 611 $100.00 CPU Cable $12.00

*Super CPU assembled with source listing but without 2K-EPROM'S (2-2708	$235.00
*Monitor in two 2708 EPROMS	$ 29.00
*CPU bare card, doc., & source	$ 59.00
*Video ram asm. 7x9 chars 64x16 line	$195.00
*Video ram bare card, doc., & source	$ 49.00
*Parallel I/O asm 100 I/O lines incl. 5 PIA's for 10 ports	$139.00
*Parallel I/O bare card & doc.	$ 49.00
*Wire-wrap/Prototype bare card	$ 39.00

BACKPLANES/MOTHERBOARDS

*16 Position SS-50	$80.00
*12 Position SS-50	$60.00
* 8 Position SS-50	$40.00
* 4 Position SS-50	$20.00
* 8 Position SS-30	$39.00

*Connectors:
Gold $1.60 ea. (M or F)
Tin M $.40 ea. F $.50 ea.

Ask about Thomas Instrumentation's Low Cost System
and the new Rack Mount Chassis

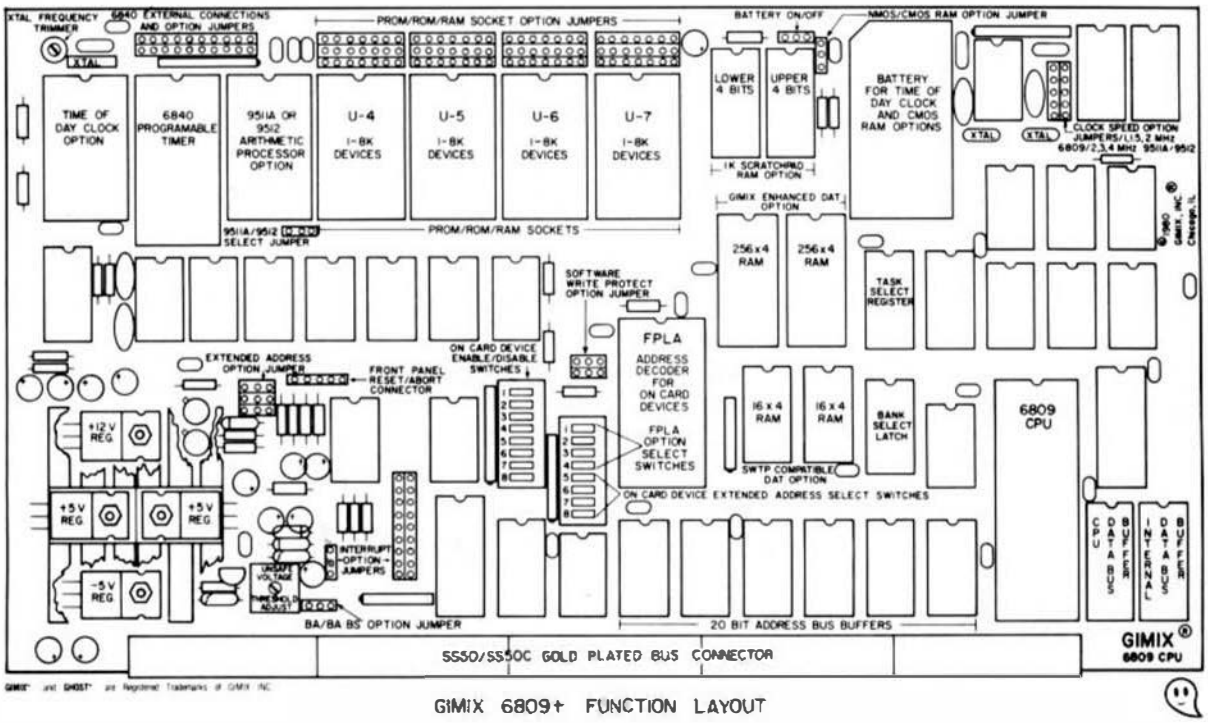
DEALERS FOR SWTPC, GIMIX, AND TSC

*All Thomas Instrumentation's cards come with full documentation including software source listings where applicable *All assembled cards are burned in at 150F and fully tested with Gold conn. *Bare card prices do not include edge connectors

THOMAS INSTRUMENTATION

168 EIGHTH STREET AVALON, NJ 08202 (609) 967-4280
NJ RES. INCLUDE 5% SALES TAX
CONT. USA INCLUDE $2.00 SHIPPING, CANADA $5.00, FOREIGN $10.00
MASTERCARD, VISA and C.O.D. ACCEPTED

GIMIX PROUDLY PRESENTS!!!



GIMIX 6809+ FUNCTION LAYOUT

SS50C 6809 CPU

The 6809 CPU card will be available in a standard version and our **6809 PLUS** version that is fully socketed to allow adding options at anytime.

- + A 6840 timer package that provides 3 independent 16 bit counters is included on all **6809 PLUS** cards.
- + A 9511 or 9512 Arithmetic Processors option with its own independent crystal that allows you to use 2, 3, or 4 MHz parts in any combination with the 6809 running at 1, 1.5, or 2 MHZ.
- + 1K of scratchpad RAM
- + A Time of Day Clock option with battery back-up. With this option you can also substitute 1K of CMOS RAM that will also be battery backed up.
- + User selectable processor speeds without having to change the crystal.

32K of PROM, ROM or RAM. Both versions have 4 sockets that can each hold from 1K to 8K parts. Single or multiple voltage parts can be used on the **PLUS** version. The standard version only allows the use of single voltage parts.

All on board devices and options can use extended addressing so that they will only respond to that page to which they are set.

The card is double buffered and allows versatility in the use of software and memory address control disciplines.

Please note that this card does not have an on board baud rate generator, and must be used in systems where baud rates (if needed) are provided elsewhere in the system.

GIMIX INC. 1337 WEST 37th PLACE • CHICAGO, IL 60609 • (312) 927-5510 • TWX 910-221-4055

GIMIX* and GHOST* are registered trademarks of GIMIX Inc. 1980 GIMIX Inc.

SS-50 MEETS UCSD PASCAL

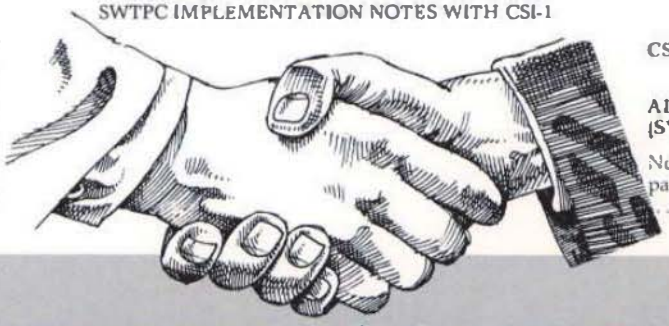
FOR 6809 SYSTEMS

IMMEDIATE DELIVERY FOR SWTPC, GIMIX USERS, 8" OR 5¼" DISKETTES
SMOKE SIGNAL BROADCASTING USERS, INQUIRE

FREE! UCSD PASCAL USERS MANUAL PLUS
SWTPC IMPLEMENTATION NOTES WITH CSI-1

CSI-1 O/S, PASCAL Compiler, Screen Editor, YALOE (Line editor for hard-copy terminals), Filer, Linker, Library, Setup, Binder, Interpreter, BIOS **$250.00**

CSI-2 BASIC Compiler, L-2 Editor, Patch, Disassembler, Calculator. **$100.00**



CSI-3 MARCO Assemblers for 6809 and 6800 ..... **$100.00**

ALL THREE DISKS AND MANUAL (SYSTEM) **$419.00**

Now Available! Pascal user-group compatible (IBM 3740) Disk Driver Routines **$20**

(SWTPC equipment compatible)

UCSD PASCAL™ MEETS BUSINESS

NEW MICRO-WINCHESTER DISK DRIVE FOR MICROCOMPUTERS

CSI announces a new option for their UDS 470 microcomputer: the micro-winchester hard disk drive. Available 4Q 1980 the micro-winchester drive will replace one of the two mini-floppy drives now standard with the UDS 470, thereby bringing a dramatic increase in on-line storage capacity.

This increased storage — 6.38 Megabytes — makes the UDS 470 viable for business applications requiring more on-line storage than previously available with floppy disk drives.

The micro-winchester drive will fit directly into the existing UDS 470 cabinet. Thus customers will realize the benefit of increased storage capacity without the disadvantage of finding space for a larger cabinet or an additional disk drive enclosure.

CALL TOLL-FREE (800) 255-4411

Continental U.S.A. only. (Kansas residents call (913) 371-6136)

CONTROL SYSTEMS, INC.
MICROSYSTEMS DIVISION

master charge
VISA
for software only

1317 Central, Kansas City, KS 66102 (913) 371-6136 • 5200 West 73rd St., Minneapolis, MN 55435 (612) 831-0214
Drawer EE, Williamsburg, VA 23185 (804) 564-9350

"UCSD Pascal" is a registered trademark of The Regents of the University of California.

'68' Micro Journal
3018 Hamlt Rd.
Hixson, TN 37343

Second Class Postage Paid
At Chattanooga, TN
ISSN 0194-5025